AFRL-IF-RS-TR-1999-227
Final Technical Report
October 1999

CONTEXT INTERCHANGE: USING KNOWLEDGE ABOUT DATA TO INTEGRATE DISPARATE SOURCES

Massachusetts Institute of Technology

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. A436

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1999-227 has been reviewed and is approved for publication.

APPROVED: Raymond A. Liuzzi

RAYMOND A. LIUZZI
Project Engineer

FOR THE DIRECTOR:

NORTHRUP FOWLER, III, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTD, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

CONTEXT INTERCHANGE: USING KNOWLEDGE ABOUT DATA TO INTEGRATE DISPARATE SOURCES

Stuart Madnick
Michael Siegel

Contractor: Massachusetts Institute of Technology
Contract Number: F30602-93-C-0160
Effective Date of Contract: 20 July 1993
Contract Expiration Date: 30 Sep 1998
Short Title of Work: Context Interchange: Using Knowledge About Data to Integrate Disparate Sources
Period of Work Covered: Jul 93 – Sep 98

Principal Investigator: Stuart Madnick
Phone: (617) 253-6671
AFRL Project Engineer: Raymond Liuzzi
Phone: (315) 330-3577

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Raymond Liuzzi, AFRL/IFTD, 525 Brooks Rd, Rome, NY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Oct 99

3. REPORT TYPE AND DATES COVERED
Final Jul 93 - Sep 98

4. TITLE AND SUBTITLE
CONTEXT INTERCHANGE: USING KNOWLEDGE ABOUT DATA TO INTEGRATE DISPARATE SOURCES

5. FUNDING NUMBERS
C - F30602-93-C-0160
PE - 62301E
PR - A436
TA - 00
WU - 01

6. AUTHOR(S)

Stuart Madnick and Michael Siegel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of Technology
Sloan School of Management
30 Wadsworth St.
Cambridge MA 02142

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFTD
525 Brooks Rd
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-1999-227

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Raymond Liuzzi, IFTD, 315-330-3577

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Context Interchange Project presents a unique approach to the problem of semantic conflict resolution among multiple heterogeneous data sources. The system presents a semantically meaningful view of the data to the receivers (e.g. user applications) for all the available data sources. The semantic conflicts are automatically detected and reconciled by a Context Mediator using the context knowledge associated with both the data sources and the data receivers. The results are collated and presented in the receiver context. The current implementation of the system provides access to flat files, classical relational databases, on-line databases, and web services.

14. SUBJECT TERMS
Database, Knowledge Base, Artificial Intelligence, Software, Computers

15. NUMBER OF PAGES
92

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

1. IMPLICATIONS FOR GOVERNMENT, BUSINESS AND INDUSTRY

Metadata Jones and the Tower of Babel:
The Challenge of Large-Scale Semantic Heterogeneity

Stuart E. Madnick

Sloan WP#4069 CISL WP #99-04

Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142

Metadata Jones and the Tower of Babel: The Challenge of Large-Scale Semantic Heterogeneity *

Stuart E. Madnick
Sloan School of Management, Massachusetts Institute of Technology
30 Wadsworth Street, Room E53-321
Cambridge, MA USA 02139
Phone: (617) 253-6671
Fax: (617) 253-3321
smadnick@mit.edu

ABSTRACT

The popularity and growth of the "Information SuperHighway" (e.g., the Web) have dramatically increased the number of information sources available for use and the opportunity for important new information-intensive applications (e.g., massive data warehouses, integrated supply chain management, global risk management, in-transit visibility). Unfortunately, there are significant challenges to be overcome regarding *data extraction* and *data interpretation* in order for this opportunity to be realized.

Data Extraction: One problem is the difficulty in easily and automatically extracting very specific data elements from Web sites for use by operational systems. New technologies, such as XML and Web Querying/Wrapping, offer possible solutions to this problem.

Data Interpretation: Another serious problem is the existence of heterogeneous contexts, whereby each SOURCE of information and potential RECEIVER of that information may operate with a different context, leading to large-scale semantic heterogeneity. A context is the collection of implicit assumptions about the context definition (i.e., meaning) and context characteristics (i.e., quality) of the information. As a simple example, whereas most US universities grade on a 4.0 scale, MIT uses a 5.0 scale – posing a problem if one is comparing student GPA's. Another typical example might be the extraction of price information from the Web: but is the price in Dollars or Yen (If dollars, is it US dollars or Hong Kong dollars), does it include taxes, does it include shipping, etc. – and does that match the receiver's assumptions?

In this paper, examples of important context challenges will be presented and the critical role of metadata, in the form of context knowledge, will be discussed.

Preamble

The Bible tells the tale of the Tower of Babel where mankind endeavored to build a tower to reach to the Heavens. According to the Bible, God introduced a multiplicity of languages – the resulting confusion made it impossible for such large-scale coordination and communication and led to the termination of the tower's construction. Today we are attempting to build "information superhighways" to access information from around the organization and around the world. Will this current great endeavor succeed or will it also be overcome by a "confusion of tongues"? The effective use of metadata can provide an approach to overcoming the challenges.

Motivation

There have been significant research efforts focused on physical information infrastructure, such as establishing high-speed data links to access information distributed throughout the world. It is increasingly obvious, however, that this kind of "physical

* This work is supported in part by DARPA and USAF/Rome Laboratory under contract F30602-93-C-0160.

connectivity" alone is not sufficient since the exchange of bits and bytes is only valuable when information can be efficiently and meaningfully exchanged. These capabilities are essential to providing the "logical connectivity" that is critically needed for dealing with the challenges of the information age.

The need for intelligent information integration is important to all information-intensive endeavors, with broad relevancy for global applications, such as Manufacturing (e.g., Integrated Supply Chain Management), Transportation/Logistics (e.g., In-Transit Visibility), Government / Military (e.g., Total Asset Visibility), Financial Services (e.g., Global Risk Management).

I. Distributed Context Knowledge to Integrate Heterogeneous Sources and Uses

Advances in computing and networking technologies now allow huge amounts of data to be gathered and shared on an unprecedented scale. Unfortunately, these new-found capabilities by themselves are only marginally useful if the information cannot be easily **extracted** and **gathered from disparate sources**, if the information is represented differently with **different interpretations**, and if it must satisfy **differing user needs**.

Some of the extraction and dissemination challenges arise because the information sources may be traditional databases, web sites, or even spreadsheets or electronic mail. Furthermore, the user may originate his or her request in a variety ways. Even more challenging to the correct interpretation of information is the fact that the sources and users may each assume different semantics or "context" (as a trivial example, one source may be assuming measurements in meters whereas another assumes feet.)

Contextual issues can be much more complex in other situations. For example, the meaning of "net sales" may vary – with "excise taxes" included for government reporting purposes in one context, but excluded for security analysis purposes in another. Also, one context may use information for a fiscal year as reported by the company, while another may use a standardized fiscal year to make all companies comparable. Furthermore, there may be multiple users that might want an answer to such a question, each with their own desired media and meaning (user context profile). Note that a "user" might be a person, an application program, a database, or a data warehouse.

In summary, to exploit the proliferation of information sources that are becoming available, we require not only technology, such as the Internet, that will provide "physical connectivity" to information sources, but also "logical connectivity" so that the information can be obtained from disparate sources and can be meaningfully assimilated. This **context knowledge** is often widely **distributed** within and across organizations. Solutions adopted to achieve interoperability must be scaleable and extensible. Thus, it is important to support **the acquisition, organization, and effective intelligent usage of distributed context knowledge**. Components of a Context Interchange System have been designed and implemented as a basic prototype at MIT.

II. The Intelligent Information Integration Challenge

Simple Example

As an illustration of the problems created by the disparities underlying the way information is provided, represented, interpreted, and used, consider the example depicted in Figure 1 below. The users wish to answer a fairly common, but important, type of question:

"**How much funds are left for project A?**" The calculation in this case is conceptually quite simple, merely subtract the expenses incurred by the 3 regions from the amount of funds that had been allocated (these are all shown on the left side under the heading labeled "Sources").

> *Although we only discuss this particular example, the reader is encouraged to consider the many other similar situations that exist in all disciplines and among all organizations.*

<u>Information Extraction and Dissemination Challenges</u>

Extraction: Even assuming that all the necessary information is available electronically and connected via the Internet, they may be in differing media and meaning. In this example, the allocated funds are in an Oracle **relational database** in Singapore, the expenses for Region 1 (USA) are available from a **web site**, the expenses for Region 2 (UK) are in an Excel **spreadsheet**, and the expenses from Region 3 (Japan) are provided via a semi-structured **electronic mail** message. In order to compute the desired answer, the information must be extracted from these varying sources and gathered together.

Dissemination: Similarly, the actual request may originate in many ways (these are shown on the right side under the heading labeled "End-User Environments & Applications"). A user in the USA may be making this request from a **Web browser**, a user in the UK may have this request originating from an "embedded SQL query" in a **spreadsheet**, a user in Singapore may be collecting this information for **data warehousing** purposes. Furthermore, this information may be requested and used as part of calculations for arbitrary **application programs** (e.g., preparation of budgeting reports, generation of exception reports, etc.)

<u>Information Interpretation Challenges</u>

Merely subtracting the numbers shown in the Figure 1 expense sources on the left from the allocated number does not produce the "right" answer because different sets of assumptions underlie the representation of the information in the sources. These assumptions are often not explicit, we call these the meaning or *context* of the information. In this case, the source contexts are indicated at the far left in Figure 1.

For the example shown in Figure 1, the allocated funds are expressed in 1000's of Singapore dollars, the expenses in Region 2 are expressed in 1's of British pounds <u>excluding</u> the 10% VAT charges, and Region 3 reports its expenses in 100's of Japanese Yen.

Likewise, the receivers' may have their own unique context, shown at the far right in Figure 1. A USA user may expect the answer in 1's of US dollars, whereas the Singapore user may wish the answer in 1000's of Singapore dollars. The UK user may want the answer in 100's of British pounds <u>including</u> the 10% VAT charges. Under these circumstances, answering even the "simple" question of Figure 1 is not so simple – try it yourself. If fact, auxiliary information sources may be needed, such as currency conversion rates, as well as rules on how such conversions should be done (e.g., as of what date).

Contextual issues can be much more complex in other situations. For example, the meaning of "net sales" may vary – with "excise taxes" included for government reporting purposes in one context, but excluded for security analysis purposes in another. Also, one context may use information for a fiscal year as reported by the company, while another may use a standardized fiscal year to make all companies comparable. Furthermore, there may be multiple users (see right side of Figure 1) that might want an answer to such a question, each with their own desired media and meaning (user context profile). Note that a "user" might be a person, an application program, a database, or a data warehouse.

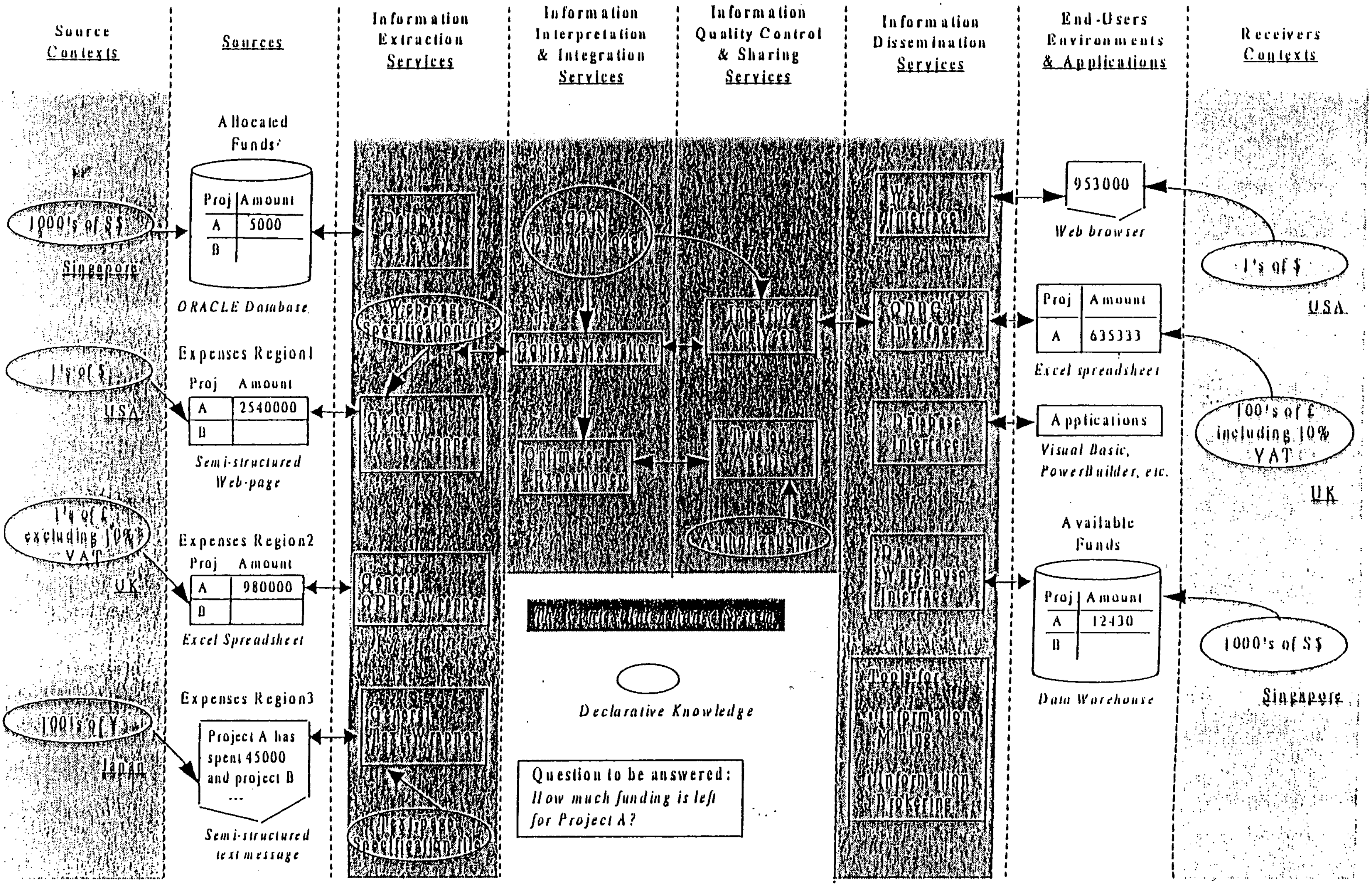


Figure 1. Example Application Illustrating Modular Architecture

In summary, it is increasingly apparent that to exploit the proliferation of information sources that are becoming available, we require not only technology, such as the Internet, that will provide "physical connectivity" to information sources, but also "logical connectivity" so that the information can be obtained from disparate sources and can be meaningfully assimilated. With the amount and diversity of information sources available it is necessary to be able to extract and organize the information from not only structured databases but also semi-structured web sources, spreadsheets, and text sources. In addition solutions adopted to achieve interoperability must be scaleable and extensible and provide decision makers with the appropriate services in an efficient and timely manner in their environments and their applications.

Basic components of a Context Interchange System, illustrated in the center portions of Figure 1, have been designed and implemented as a limited prototype. In one sample application, it makes use of several online databases (e.g., Disclosure, Worldscope, Datastream – historical financial information sources), various web sites (e.g., Security APL – current stock exchange prices, Edgar – USA SEC filings, and Olsen – currency conversion information), and semi-structured documents (e.g., Merrill Lynch analyst reports). The financial information needed to answer a question are extracted from these sources, correctly interpreted (involving automatic conversions), integrated and disseminated in various ways, such as into an Excel spreadsheet application of a financial analyst.

III. Overview of the Context Interchange Approach

1. Context Interchange Architecture.

Context Interchange is a mediation approach for semantic integration of disparate (heterogeneous and distributed) information sources. It has been described in [GBMS96a]. The Context Interchange approach includes not only the mediation infrastructure and services, but also wrapping technology and middleware services for accessing the source information and facilitating the integration of the mediated results into end-users applications.

The architecture comprises three categories of components: the wrappers, the mediation services, and the middleware, interface, and facilitation services.

The wrappers are physical and logical gateways providing a uniform access to the disparate sources over the network.

The set of Context Mediation Services, comprises a Context Mediator, a Query Optimizer and a Query Executioner. The Context Mediator is in charge of the identification and resolution of potential semantic conflicts induced by a query. This automatic detection and reconciliation of conflicts present in different information sources is made possible by general knowledge of the underlying application domain, as well as informational content and implicit assumptions associated to the receivers and sources. These bodies of declarative knowledge are represented in the form of a domain model, a set of elevation axioms, and a set of context theories respectively.

The result of the mediation is a mediated query. To retrieve the data from the disparate information sources, the mediated query is then transformed into a query execution plan, which is optimized, taking into account the topology of the network of sources and their capabilities. The plan is then executed to retrieve the data from the various sources, results are composed as a message, and sent to the receiver.

The middleware, interface, and facilitation services are the services which give access to the mediation services for users and application programs. They rely on an Application Programming interface and a protocol implemented as a standard subset of the Open Data Base Connectivity (ODBC) protocol tunneled into the HyperText Transfer Protocol (HTTP). Examples of interfaces and facilitation services are the Query-By-Example Web interface which is a *point-and-click* interface for the construction of ad-hoc queries [Jako96], and the Context ODBC driver [Shum96] which gives access to the mediation infrastructure to any ODBC-compliant Windows 95 or Windows NT application (Excel, Access, etc.).

2. Wrapping.

Wrappers serve as gateways to external information sources for mediation services engines. While information sources vary widely in interface technology and physical data representation, the wrappers should provide a uniform interface to the sources. Two general classes of information sources are: structured data sources, such as traditional relational DBMS's (Oracle and Ingres), and on-line information services, such as Web sites reached though navigable HyperText Markup Language (HTML) pages.

COIN wrappers [Qu96] present a common client interface with the appearance of a relational table to the mediation services engine. The protocol used at the wrapper interface is identical to the protocol for accessing mediation services – ODBC tunneled into HyperText Transfer Protocol (HTTP). Requests are presented in SQL. Results are returned in the form of standard objects, such as HTML tables or JavaScript objects. Because of the common interface at each stage, a user can, in fact, by-pass mediation services and directly access raw data from a source through a wrapper.

COIN wrappers for relational DBMS's serve as protocol converters. Queries or other access requests are received from the client in COIN protocol. The SQL is extracted and presented to the DBMS using its own API. Query results are then obtained from the DBMS API and delivered to the client using the COIN protocol via HTTP.

For the Web sources, we have developed a generic Web-wrapping technology, which is capable of extracting semi-structured information from Web-services. The COIN Web-wrapping technology is unique for it takes advantage of the Hypertext structure of Web-sources and of the underlying structure provided by the HTML. We treat a Web service as a collection of static and dynamic pages connected by transitions.

Information on the Web is often not contained on a single page, but is distributed over a group of pages linked by static (e.g. <A HREF=...>) and dynamic hypertext links (e.g. <FORM ACTION=...>). In fact, whether a "service" is located on a single Web-server, or distributed over a number of independently maintained sites, is transparent to the user. Typically, a user may contact the "home page" of the service, click on hypertext links, retrieve some information, fill in and post HTML forms, obtain another piece of information, and so on. The various pieces of information located on one page can be: in a pre-structured format, in a semi-structured format, or in unstructured plain text.

By pre-structured format, we mean a format which is known in advance by the user. This is typically the case of pages using a data representation compliant with a standard, such as the Open Financial Connectivity standard. Where information producers are able to agree on such standard representations COIN can take advantage of the format guarantees.

Semi-structured format includes data presented in a table, a list, a tree, or other structuring organization, but for which the structure is not fully know in advance and must be parsed and analyzed on the fly to locate the data. There are a large number of information sources on the World Wide Web (e.g., CIA fact book, stock exchange quote services, weather reports and weather forecasts, etc.) offering semi-structured format data.

The COIN Web-wrapping technology is based on a high level declarative language for the specification of wrapper interface and actions. This language specifies what information can be extracted from a source. The generic wrapper engine transforms user requests into a plan for extracting the relevant data according to the specification, executes the plan by accessing the source, and organizes and presents the extracted data. The specification language for the generic Web-wrappers allows the definition of a state transition network. The transitions in the network correspond to the hyperlinks in the hypertext, additionally, the information initially inputted or collected in the preceding stages is carried and is used to define the transitions, fill the parameters of a form, or choose a link among several on a page. On each page (or state of the transition network), the Web-wrapper specification uses patterns (e.g., regular expressions) to identify the location of data to be extracted, input fields for a form, and links to other locations. More recently we have moved beyond regular expression patterns so that we can take advantage of the structure of information on a page as provided by XML tags.

Furthermore, web sites have differing capabilities. Some sites are collections of static pages, others are dynamically created pages based upon specific interactions. It is necessary to take into consideration the specific capabilities and limitations on data retrieval from sites.

3. Mediation.

In a heterogeneous and distributed environment, the mediator transforms a query written in the terms known to the user or application program (i.e., according to the user's or programmer's assumptions and knowledge) into one or more queries in the terms of the component sources. The individual subqueries may still involve several sources. Subsequent planning, optimization and execution phases are needed. Typically, the planning and execution phases will consider the limitations of the sources and the topology and costs of the network. The execution phase is in charge of the scheduling of the query execution plan and the realization of the complementary operations that could not be handled by the sources individually (e.g. a join across sources).

The first mediation phase can be naively described as the rewriting of the query against a "view definition", the view of the disparate information sources that the mediation service provides to the user or application program. The main quality of the mediation approach will depend on its properties with respect to the strategy for the assimilation and definition of the knowledge needed for the construction of this "view definition." Where a large number of independent information sources are accessed (as is now possible with the global information infrastructure), flexibility, scaleability, and non-intrusiveness will be of primary importance.

Traditional tight-coupling approaches to semantic interoperability rely on the *a priori* creation of federated views on the heterogeneous information sources. Although they provide good support for data access, they do not scale-up efficiently given the complexity involved in constructing and maintaining a shared schema for a large number of, possibly independently managed and evolving, sources. Loose-coupling approaches rely on the user's intimate knowledge of the semantic conflicts between the sources and the conflict resolution procedures.

This flexibility becomes a drawback for scaleability when this knowledge grows and changes as more sources are join the system and when sources are changing.

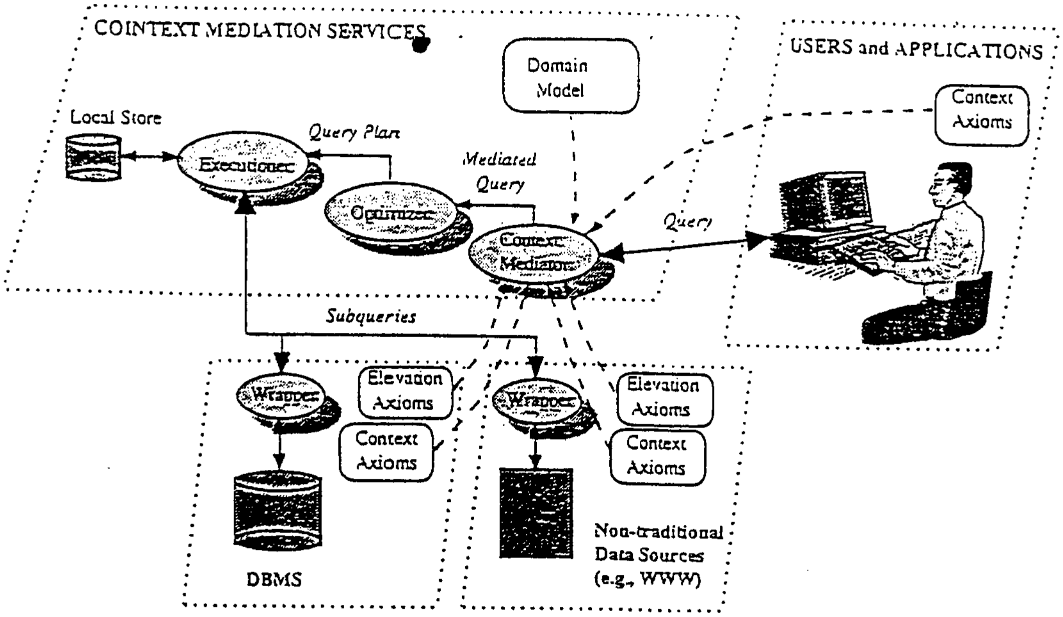


Figure 2. The Architecture of the Context Interchange System

The Context Interchange (COIN) approach is a middle ground between these two approaches. It allows queries to the sources to be mediated, i.e. semantic conflicts to be identified and solved by a context mediator through comparison of contexts associated with the sources and receivers concerned by the queries. It only requires the minimum adoption of a common Domain Model which defines the domain of discourse of the application.

The knowledge needed for integration is formally modeled in a COIN framework [Goh96], The COIN framework is a mathematical structure offering a sound foundation for the realization of the Context Interchange strategy. The COIN framework comprises a data model and a language, called COINL, of the Frame-Logic (F-Logic) family [KLW95, DoT95]. The framework is used to define the different elements needed to implement the strategy in a given application:

- The Domain Model is a collection of rich types (semantic types) defining the domain of discourse for the integration strategy;
- Elevation Axioms for each source identify the semantic objects (instances of semantic types) corresponding to source data elements and define integrity constraints specifying general properties of the sources;
- Context Definitions define the different interpretations of the semantic objects in the different sources or from a receiver's point of view.

The Domain Model, the different sets of Elevation Axioms, the Context Definitions, together with additional generic axioms defining the mediation, constitute a COINL program. This program controls the query mediation engine.

Let us consider a simple example where a user issues the query Q1 to a source called "security" providing historical financial data about a stock exchange. The user and the source have different assumptions regarding the interpretation of the data. These assumptions are captured in their respective contexts C1 and C2. The Domain Model defines semantic types such as money amounts, dates, and company identifications. Query Q1 requests the price of the IBM security on March 12, 1995:

```
(Q1)   select security.Price
                from security
                where security.Ticker = "IBM"
               and security.Date = "12/03/95";
```

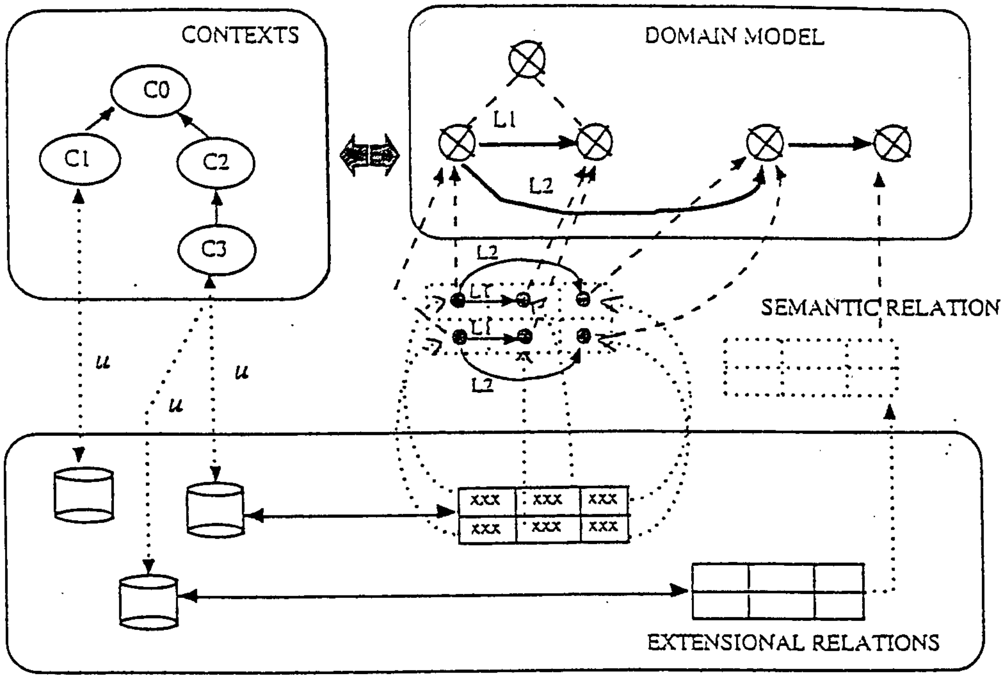


Figure 3. The Context Interchange Formal Framework

The receiver's context C1 assumes money amounts are in French Francs, dates in the European format, and that currency conversions should use the date of the money amount validity. We see immediately that context information is needed to avoid the confusion between March 12 and December 3, 1995. On the other hand, the source context C2 expresses its money amounts in the local currencies of the company, and dates are in American format. The mediation rewrites the query, incorporating the proper currency conversion (as of March 12, 1995) making use of an ancillary source called "cc" for exchange rates, and the proper date format conversion. The resulting mediated query MQ1 is:

```
(MQ1) select security.Price * cc.Rate
               from security, cc
               where security.Ticker = "IBM"
                 and security.Date = "03/12/95"
                 and cc.Expressed = "USD"
                 and cc.Exchanged = "FRF"
                 and cc.AsOfDay = security.Date;
```

In this example, the domain model will define the various semantic types corresponding to the concepts associated to the data elements manipulated in the application domain. For instance, semantic types capturing notions like money amounts, company financials or exchange rates need to be defined. If some relationships exist among these semantic types and are relevant from an ontological point of view (as opposed to the peculiarities of the structures hosting the data in the sources), they can be represented in the domain model by means of attributes. The following is an excerpt of the domain model of our example in COINL[1] :

```
moneyAmount: number;
companyFinancials: moneyAmount;
exchangeRate: number [     to ⇒ currency;
                           from ⇒ currency;
                           asof ⇒ date].
```

The elevation axioms define the semantic image of the relations and the data exported by the sources. Below is an excerpt of the elevation axioms for a source exporting a relation **Olsen** reporting historical data for currency exchange rates (Olsen is an actual Web site, which can be utilized as if it were a relational database through use of COIN's Web Wrapping technology). The first rule defines the semantic relation **Olsen_semantic.** The second rule, defines an exchangeRate semantic object. The third rule is an integrity constraint expressing the reversibility of the rate.

```
Olsen_semantic(     f_to(To, From, Date),
                    f_from(To, From, Date),
                    f_date(To, From, Date),
                    f_rate(To,From, Date)) ←
                                 olsen(To, From, Date, Rate).
f_rate(To, From, Date): exchangeRate
               [to ⇒ f_to(To, From, Date),
                from ⇒ f_from(To, From, Date),
                date ⇒ f_date(To, From, Date)].
Olsen(To, From, Date, Rate1), Olsen(From, To, Date, Rate2)→
                              Rate1 = 1/Rate2.
```

The context associated with the sources and the receivers define the *modifiers* of the semantic objects. The modifiers are special attributes dependent on the context and determine the interpretation of the data. They are used for the identification of conflicts during the query mediation. They can be defined by extension (given a value) or by intention (by means of a rule). Several modifiers corresponding to different notions determining the interpretation of a semantic object are associated to it (e.g., the currency and the as-of date of a money amount). Modifiers are declared for all objects of a given semantic type.

```
X:moneyAmount
     [[currency.value ⇒ "FRF"];
      [asofdate ⇒ V] → X[report.date ⇒ V] ].
```

Finally, the conversion functions for each modifier locally defines the resolution of potential conflicts. The conversion functions can be defined in COINL but are likely, in practical cases, to rely on external services or external procedures. The relevant conversion functions are

[1] In this document we are using the abstract syntax of COINL in order to give the reader an intuition of the logical constructs in the language. End-users and programmers are offered visual or graphical interfaces and a concise concrete syntax (of the family of OQL).

gathered and composed during mediation to resolve the conflicts. No global or exhaustive pairwise definition of the conflict resolution procedures is needed.

Both the query to be mediated and the COINL program are combined into a definite logic program (a set of Horn clauses) where the translation of the query is a goal. The mediation is performed by an abductive procedure which infers from the query and the COINL programs a reformulation of the initial query in the terms of the component sources. The abductive procedure makes use of the integrity constraints in a constraint propagation phase which has the effect of a semantic query optimization. For instance, logically inconsistent rewritten queries are rejected, rewritten queries containing redundant information are simplified, rewritten queries are augmented with auxiliary information.

Although the procedure itself is inspired by the Abductive Logic Programming framework [KKT93] and can be qualified as an abduction procedure, we do not argue that abduction by itself is a suitable philosophical concept for mediation, but rather take advantage of formal logical framework for the study and implementation of an appropriate procedure. One of the main advantages of the abductive logic programming framework is the simplicity in which it can be used to formally combine and to implement features of query processing, semantic query optimization and constraint programming.

The COIN abductive framework can also be extrapolated to problem areas such as integrity management, view updates and intensional updates for databases. Because of the clear separation between the declarative definition of the logic of mediation into the COINL program from the generic abductive procedure for query mediation, we are able to adapt our mediation procedure to new situations such as mediated consistency management across disparate sources, mediated update management of one or more database using heterogeneous external auxiliary information or mediated monitoring of changes. Although there are fundamental theoretical limits in many areas, such as view update, we can extend the range of mediation services to handle a broader range of client needs.

The mediated update problem illustrates the potential advantage of the formal logical approach in COIN over traditional view mechanisms for mediation. For a retrieval, either approach can be made to deliver correct results (with more or less effort). The COIN approach, however, holds the knowledge of the semantics of data in each context and across contexts in declarative logical statements separate from the mediation procedure. An update asserts that certain data objects must be made to have certain values in the updater's context. An update mediation algorithm by combining the update assertions with the COIN logical formulation of context semantics, can determine whether is unambiguous and feasible, and if so, what source data updates must be made to achieve the intended results. If ambiguous or otherwise infeasible, the logical representation may be able to indicate what additional constraints would clarify the updater's intention sufficiently to the update to proceed.

We are also applying the COIN framework to important aspects of the source selection problem. Integrity constraints in COINL and the consistency checking component of the abductive procedure provide the basic ingredients to characterize the scope of information available from each source, to efficiently rule out irrelevant data sources and thereby speed up the selection process. For example, a query requesting information about *companies with assets lower than $2 million* can avoid accessing a particular source based on knowledge of integrity constraints stating that *the source only reports information about companies listed in the New York Stock Exchange* (NYSE), and that *companies must have assets larger than $10 million to be listed in the NYSE*. In general, integrity constraints express necessary conditions imposed on data. However, more generally, a notion of completeness degree of the domain of the source with respect to the constraint captures a richer semantic information and allows more powerful source selection. For instance, a source could contain exactly or at least all the data verifying the constraint (information about all the companies listed in the NYSE are exhaustively reported in the source).

Conclusion

We are in the midst of exciting times – the opportunities to make use of diverse information sources are incredible but the challenges are considerable. The effective use of metadata can enable us to overcome the challenges and more fully realize the opportunities. A particularly interesting aspect of the context mediation approach described is the use of metadata to describe the expectations of the receiver as well as the semantics assumed by the sources. If we do not address these challenges directly and effectively, we might endure serious consequences, as illustrated by the historical example displayed in the box below.

The 1805 Overture

In 1805, the Austrian and Russian Emperors agreed to join forces against Napoleon. The Russians promised that their forces would be in the field in Bavaria by **Oct. 20**.

The Austrian staff planned its campaign based on that date in the **Gregorian calendar**. Russia, however, still used the ancient **Julian calendar**, which lagged 10 days behind.

The calendar difference allowed Napoleon to surround Austrian General Mack's army at Ulm and force its surrender on Oct. 21, well before the Russian forces could reach him, ultimately setting the stage for Austerlitz.

Source: David Chandler, *The Campaigns of Napoleon*, New York: MacMillan 1966, pg. 390.

ACKNOWLEDGEMENTS

Work reported herein has been supported, in part, by the Advanced Research Projects Agency (ARPA) and the USAF/Rome Laboratory under contract F30602-93-C-0160, the MIT International Financial Services Research Center (IFSRC), the MIT Leaders For Manufacturing (LFM) Program, the MIT PROductivity From Information Technology (PROFIT) Program, and the MIT Total Data Quality Management (TDQM) Program. Information about the Context Interchange project can be obtained at http://context.mit.edu.

REFERENCES

[BFG96a] Bressan, S., Fynn, K., Goh, C.H., Jakobisiak, M., Hussein, K., Kon, H., Lee, T., Madnick, S., Pena, T., Qu, J., Shum, A., and Siegel, M. (1996). "The COntext INterchange mediator prototype," *SIGMOD97*.

[BFG96b] Bressan, S., Fynn, K., Goh, C.H., Madnick, S., Pena, T., and Siegel, M. (1996). "Overview of a prolog implementation of the COntext INterchange mediator," *Practical Applications of Prolog 97*.

[DaGH95] Daruwala, A., Goh, C.H., Hofmeister, S., Hussein, K., Madnick, S., and Siegel, M. (1995). "The context interchange network prototype," In *Proc of the IFIP WG2.6 Sixth Working Conference on Database Semantics (DS-6)*, Atlanta, GA. To appear in LNCS (Springer-Verlag).

[DoT95] Dobbie, G. and Topor, R. (1995). "On the declarative and procedural semantics of deductive object-oriented systems," *Journal of Intelligent Information Systems*, 4:193--219.

[Goh96] Goh, C. (1996). *Representing and Reasoning about Semantic Conflicts* In *Heterogeneous Information System*, PhD Thesis, MIT.

[GBMS96a] Goh, C.H., Bressan, S., Madnick, S.E., and Siegel, M.D. (1996). "Context Interchange: Representing and reasoning about data semantics in heterogeneous systems," *Sloan School Working Paper #3928*, Sloan School of Management, MIT, 50 Memorial Drive, Cambridge MA 02139.

[GBMS96b] Goh, C.H., Bressan, S., Madnick, S.E., and Siegel, M.D. (1996). "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," *Sloan School Working Paper*, Sloan School of Management, MIT, 50 Memorial Drive, Cambridge MA 02139.

[GMS94] Goh, C.H., Madnick, S.E., and Siegel, M.D. (1994). "Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment," In *Proc of the Third International Conference on Information and Knowledge Management*, pages 337--346, Gaithersburg, MD.

[Jako96] Jakobisiak, M. (1996). "Programming the web -- design and implementation of a multidatabase browser," *Technical Report CISL WP #96-04*, Sloan School of Management, Massachusetts Institute of Technology.

[KKT93] Kakas, A.C., Kowalski, R.A., and Toni, F. (1993). "Abductive logic programming," *Journal of Logic and Computation*, 2(6):719--770.

[KLW95] Kifer, M., Lausen, G., and Wu, J. (1995). "Logical foundations of object-oriented and frame-based languages," *JACM*, 4:741—843.

[Qu96] Qu, J.F. (1996). "Data wrapping on the world wide web," *Technical Report CISL WP #96-05*, Sloan School of Management, Massachusetts Institute of Technology.

[ScSR94] Sciore, E., Siegel, M., and Rosenthal, A. (1994). "Using semantic values to facilitate interoperability among heterogeneous information systems," *ACM Transactions on Database Systems*, 19(2):254--290.

[Shum96] Shum, A. (1996). *Open Database Connectivity of the Context Interchange System*, Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

[SM91a] Siegel, M. and Madnick, S. (1991). "Context Interchange: Sharing the Meaning of Data," *SIGMOD RECORD*, Vol. 20, No. 4, December p. 77-8.

[SM91b] Siegel, M. and Madnick, S. (1991). "A metadata approach to solving semantic conflicts," In *Proc of the 17th International Conference on Very Large Data Bases*, pages 133--145.

2. CONTEXT INTERCHANGE THEORY

Context Interchange: New Features and Formalisms for the Intelligent Integration of Information

C. H. Goh, S. Bressan, S. Madnick, M. Siegel

CISL WP#97-03
February 1997
(Revised August 1999)

Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142

Context Interchange: New Features and Formalisms for the Intelligent Integration of Information

CHENG HIAN GOH
National University of Singapore
and
STÉPHANE BRESSAN, STUART MADNICK, and MICHAEL SIEGEL
Massachusetts Institute of Technology

The *Context Interchange strategy* presents a novel perspective for mediated data access in which semantic conflicts among heterogeneous systems are not identified a priori, but are detected and reconciled by a *context mediator* through comparison of *contexts axioms* corresponding to the systems engaged in data exchange. In this article, we show that queries formulated on shared views, export schema, and shared "ontologies" can be mediated in the same way using the *Context Interchange framework*. The proposed framework provides a logic-based object-oriented formalism for representing and reasoning about data semantics in disparate systems, and has been validated in a prototype implementation providing mediated data access to both traditional and web-based information sources.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Systems—*Query processing*; H.2.5 [**Database Management**]: Heterogeneous Databases—*Data translation*

General Terms: Design

Additional Key Words and Phrases: Abductive reasoning, information integration, mediators, semantic heterogeneity, semantic interoperability

We dedicate this work to the memory of Cheng Hian Goh (1965–1999).

This work is supported in part by ARPA and USAF/Rome Laboratory under contract F30602-93-C-0160, the International Financial Services Research Center (IFSRC), the PROductivity From Information Technology (PROFIT) project at MIT, and TASC, Inc., as well as NUS RP970628.

Authors' addresses: S. Bressan, S. Madnick, and M. Siegel, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139; email: sbressan@mit.edu; smadnick@mit.edu; msiegel@mit.edu.

ACM Transactions on Information Systems, Vol. 17, No. 3, July 1999, Pages 270–293.

1. INTRODUCTION

The number of online information sources and receivers has grown at an unprecedented rate in the last few years, contributed in large part by the exponential growth of the Internet as well as advances in telecommunications technologies. Nonetheless, this increased *physical connectivity* (the ability to exchange bits and bytes) does not necessarily lead to *logical connectivity* (the ability to exchange information meaningfully). This problem is sometimes referred to as the need for *semantic interoperability* [Sheth and Larson 1990] among *autonomous* and *heterogeneous* systems.

The *Context Interchange strategy* [Siegel and Madnick 1991; Sciore et al. 1994] is a mediator-based approach [Wiederhold 1992] for achieving semantic interoperability among heterogeneous sources *and* receivers, constructed on the following tenets:

—the *detection* and *reconciliation* of semantic conflicts are system services which are provided by a *Context Mediator*, and should be transparent to a user; and

—the provision of such a mediation service requires only that the user furnish a logical (declarative) specification of *how data are interpreted* in sources and receivers, and *how conflicts, when detected, should be resolved*, but *not what conflicts exists a priori* between any two systems.

This approach toward semantic interoperability is unlike most traditional integration strategies which either require users to engage in the detection and reconciliation of conflicts (in the case of *loosely coupled systems*, e.g., MRDSM [Litwin and Abdellatif 1987], VIP-MDBMS [Kuhn and Ludwig 1988]), or insist that conflicts should be identified and reconciled, a priori, by some system administrator, in one or more shared schemas (as in *tightly coupled systems*, e.g., Multibase [Landers and Rosenberg 1982] and Mermaid [Templeton et al. 1987]). In addition, the proposed framework plays a complementary role to an emerging class of integration strategies [Levy et al. 1995b; Ullman 1997] where queries are formulated on an "ontology" without specifying a priori what information sources are relevant for the query. Although the use of a logical formalism for information integration is not new (see, for example, Catarci and Lenzerini [1993] where interschema dependencies are represented using *description logics*), our integration approach is different because we have chosen to focus on the semantics of individual data items as opposed to conflicts at the schematic level.

With the above observations in mind, our goal in this article is (1) to illustrate various novel features of the Context Interchange mediation strategy and (2) to describe how the underlying representation and reasoning can be accomplished within a formal *logical* framework. Even though this work originated from a long-standing research program, the features and formalisms presented in this article are new with respect to our previous works. Our proposal is also capable of supporting "multidatabase" queries, queries on "shared views," as well as queries on shared "ontologies," while allowing semantic descriptions of disparate sources to remain loosely coupled to one another. The feasibility of this work has also been validated in a prototype system which provides access to both traditional data sources (e.g., Oracle data systems) as well as semistructured information sources (e.g., Web sites).

The rest of this article is organized as follows. Following this introduction, we present a motivational example which is used to highlight the Context Interchange strategy. Section 3 describes the Context Interchange framework by introducing both the representational formalism and the logical inferences underlying query mediation. Section 4 compares the Context Interchange strategy with other integration approaches which have been reported in the literature. The last section presents a summary of our contributions and describes some ongoing thrusts.

Due to space constraints, we have aimed at providing the intuition by grounding the discussion in examples where possible; a substantively longer version of the article, presenting more of the technical details, is available as a working paper [Goh et al. 1996]. A report on the Prototype can also be found in Bressan et al. [1997a]. An in-depth discussion of the context mediation procedure can be found in a separate report [Bressan et al. 1997b].

As one might easily gather from examining the literature, research in information integration is making progress in leaps and bounds. A detailed discussion on the full variety of integration approaches and their accomplishments is beyond the scope of this article, and we gladly recommend Hull [1997] for a comprehensive survey.

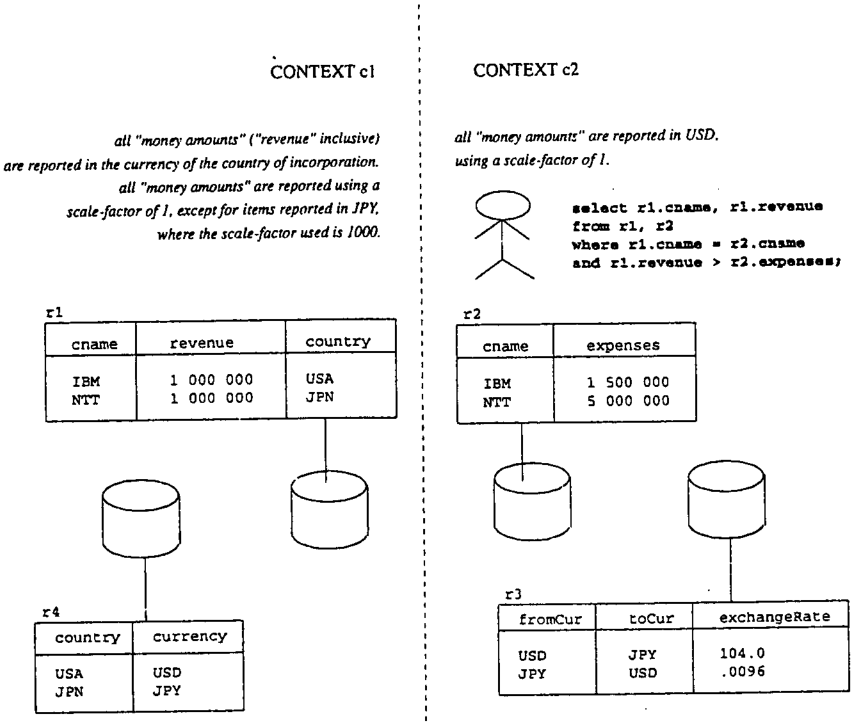
2. CONTEXT INTERCHANGE BY EXAMPLE

Consider the scenario shown in Figure 1, deliberately kept simple for didactical reasons. Data on "revenue" and "expenses" (respectively) for some collection of companies are available in two autonomously administered data sources, each comprised of a single relation denoted by r1 and r2 respectively. Suppose a user is interested in knowing which companies have been "profitable" and their respective revenue: this query can be formulated directly on the (export) schemas of the two sources as follows:

```
Q1: SELECT r1.cname, r1.revenue FROM r1, r2
    WHERE r1.cname = r2.cname AND r1.revenue > r2.expenses;
```

(We assume, without loss of generality, that relation names are unique across all data sources. This can always be accomplished via some renaming scheme: say, by prefixing the relation name with the name of the data source (e.g., db1#r1).) In the absence of any mediation, this query will return the empty answer if it is executed over the extensional data set shown in Figure 1.

The above query, however, does not take into account the fact that both sources and receivers may have different *contexts*, i.e., they may embody different assumptions on how information present should be interpreted. To simplify the ensuing discussion, we assume that the data reported in the two sources differ only in the currencies and scale-factors of "money


CONTEXT c1
all "money amounts" ("revenue" inclusive)
are reported in the currency of the country of incorporation.
all "money amounts" are reported using a
scale-factor of 1, except for items reported in JPY,
where the scale-factor used is 1000.
CONTEXT c2
all "money amounts" are reported in USD.
using a scale-factor of 1.
r1
r2
r3
r4

```
select r1.cname, r1.revenue
from r1, r2
where r1.cname = r2.cname
and r1.revenue > r2.expenses;
```

cname	revenue	country
IBM	1 000 000	USA
NTT	1 000 000	JPN

cname	expenses
IBM	1 500 000
NTT	5 000 000

fromCur	toCur	exchangeRate
USD	JPY	104.0
JPY	USD	.0096

country	currency
USA	USD
JPN	JPY

Fig. 1. Example scenario.

amounts." Specifically, in Source 1, all "money amounts" are reported using a scale-factor of 1 and the currency of the country in which the company is "incorporated"; the only exception is when they are reported in Japanese Yen (JPY); in which case the scale-factor is 1000. Source 2, on the other hand, reports all "money amounts" in USD using a scale-factor of 1. In the light of these remarks, the (empty) answer returned by executing Q1 is clearly not a "correct" answer, since the revenue of NTT (9,600,000 USD = 1,000,000 × 1,000 × 0.0096) is numerically larger than the expenses (5,000,000) reported in r2. Notice that the derivation of this answer requires access to other sources (r3 and r4) not explicitly named in the user query.

In a Context Interchange system, the semantics of data (of those present in a source, or of those expected by a receiver) can be explicitly represented in the form of a *context theory* and a set of *elevation axioms* with reference to a *domain model* (more about these later). As shown in Figure 2, queries submitted to the system are intercepted by a *Context Mediator*, which rewrites the user query to a *mediated query*. The *Optimizer* transforms this to an optimized query plan, which takes into account a variety of cost information. The optimized query plan is executed by an *Executioner* which dispatches subqueries to individual systems, collates the results, undertakes conversions which may be necessary when data are exchanged between two systems, and returns the answers to the receiver. In the

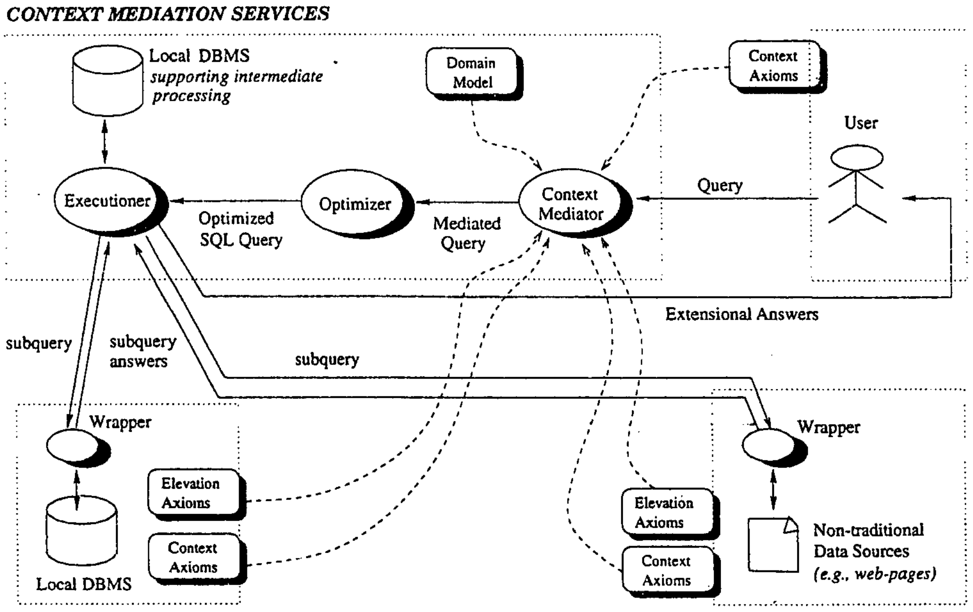


Fig. 2. Architecture of a Context Interchange system.

remainder of this section, we describe three different paradigms for supporting data access using this architecture.

2.1 Mediation of "Multidatabase" Queries

The query Q1 shown above is in fact similar to "multidatabase" MDSL queries described in Litwin and Abdellatif [1987] whereby the export schemas of individual data sources are explicitly referenced. Nonetheless, unlike the approach advocated in Litwin and Abdellatif [1987], users remain insulated from underlying semantic heterogeneity, i.e., they are not required to undertake the detection or reconciliation of potential conflicts between any two systems. In the Context Interchange system, this function is assumed by the Context Mediator: for instance, the query Q1 is transformed to the mediated query MQ1:

```
MQ1: SELECT r1.cname, r1.revenue FROM r1, r2, r4
     WHERE r1.country = r4.country
     AND r4.currency = 'USD'
     AND r1.cname = r2.cname
     AND r1.revenue > r2.expenses;
     UNION
     SELECT r1.cname, r1.revenue * 1000 * r3.rate
     FROM r1, r2, r3, r4
     WHERE r1.country = r4.country
     AND r4.currency = 'JPY'
     AND r1.cname = r2.cname
     AND r3.fromCur = 'JPY'
     AND r3.toCur = 'USD'
     AND r1.revenue * 1000 * r3.rate > r2.expenses
```

```
UNION
SELECT r1.cname, r1.revenue * r3.rate
FROM r1, r2, r3, r4
WHERE r1.country = r4.country
AND r4.currency <> 'USD'
AND r4.currency <> 'JPY'
AND r3.fromCur = r4.currency
AND r3.toCur = 'USD'
AND r1.cname = r2.cname
AND r1.revenue * r3.rate > r2.expenses;
```

This mediated query considers all potential conflicts between relations r1 and r2 when comparing values of "revenue" and "expenses" as reported in the two different contexts. Moreover, the answers returned may be further transformed so that they conform to the context of the receiver. Thus in our example, the revenue of NTT will be reported as 9 600 000 as opposed to 1 000 000. More specifically, the three-part query shown above can be understood as follows. The first subquery takes care of tuples for which revenue is reported in USD using scale-factor 1; in this case, there is no conflict. The second subquery handles tuples for which revenue is reported in JPY, implying a scale-factor of 1000. Finally, the last subquery considers the case where the currency is neither JPY nor USD, in which case only currency conversion is needed. Conversion among different currencies is aided by the ancillary data sources r3 (which provides currency conversion rates) and r4 (which identifies the currency in use corresponding to a given country). The mediated query MQ1, when executed, returns the "correct" answer consisting only of the tuple ⟨'NTT', 9 600 000⟩.

2.2 Mediation of Queries on "Shared Views"

Although "multidatabase" queries may provide users with greater flexibility in formulating a query, they also require users to know what data are present *in which data sources* and be sufficiently familiar with the attributes in different schemas (so as to construct a query). An alternative advocated in the literature is to allow *views* to be defined on the source schemas and have users formulate queries based on the view instead. For example, we might define a view on relations r1 and r2, given by

```
CREATE VIEW v1 (cname, profit) AS
SELECT r1.cname, r1.revenue - r2.expenses
FROM r1, r2
WHERE r1.cname = r2.cname;
```

in which case, query Q1 can be equivalently formulated on the view v1 as

```
VQ1: SELECT cname, profit FROM v1
     WHERE profit > 0;
```

While this view approach achieves essentially the same functionalities as tightly coupled systems, *notice that view definitions in our case are no longer concerned with semantic heterogeneity and make no attempts at identifying or resolving conflicts, since query mediation can be undertaken* *by the Context Mediator as before.* Specifically, queries formulated on the shared view can be easily rewritten to queries referencing sources directly, which allows it to undergo further transformation by the Context Mediator as before.

2.3 Mediation of Queries on Shared "Ontologies"

Yet another approach for achieving read-only integration is to define a shared domain model (often called an *ontology*), which serves as a global schema identifying all information relevant to a particular application domain. However, unlike the traditional tight-coupling approach, data held in the source databases is expressed as views over this global schema [Levy et al. 1995b; Ullman 1997]. This means that queries formulated on the ontology must be transformed to "equivalent" queries on actual data sources.

It is important to note that current work in this direction has been largely focused on designing algorithms for realizing query rewriting with the goal of identifying the relevant information sources that must be accessed to answer a query (see, for example, Levy et al. [1995a] and Ullman [1997]). In all instances that we know of, it is assumed that no semantic conflicts whatsoever exist among the disparate data sources. It should be clear that the work reported here complements rather than competes with this "new wave" integration strategy.

3. THE CONTEXT INTERCHANGE FRAMEWORK

McCarthy [1987] points out that statements about the world are never always true or false: the truth or falsity of a statement can only be understood with reference to a given *context*. This is formalized using assertions of the form

$$\bar{c}: \quad ist(c, \sigma)$$

which suggests that the statement σ is true in ("*ist*") the context c, this statement itself being asserted in an *outer context* $\bar{c}$.

McCarthy's notion of "contexts" provides a useful framework for modeling statements in heterogeneous databases which are seemingly in conflict with one another: specifically, factual statements present in a data source are not "universal" facts about the world, but are true relative to the context associated with the source but not necessarily so in a different context. Thus, if we assign the labels c1 and c2 to contexts associated with sources 1 and 2 in Figure 1, we may now write

$\bar{c}$: *ist*(c1,r1("NTT", 1 000 000, "JPN")).
$\bar{c}$: *ist*(c2, r2("NTT", 5 000 000)).

where $\bar{c}$ refers to the ubiquitous context associated with the integration exercise. For simplicity, we will omit $\bar{c}$ in the subsequent discussion, since the context for performing this integration remains invariant.

The Context Interchange framework constitutes a formal, logical specification of the components of a Context Interchange system. This comprises three components:

—The *domain model* is a collection of "rich" types, called *semantic types*, which defines the application domain (e.g., medical diagnosis, financial analysis) corresponding to the data sources which are to be integrated.

—The *elevation axioms* corresponding to each source identify the correspondences between attributes in the source and semantic types in the domain model. In addition, it codifies the integrity constraints pertaining to the source; although the integrity constraints are not needed for identifying sound transformations on user queries, they are useful for simplifying the underlying representation and for producing queries which are more optimal.

—The *context axioms*, corresponding to named contexts associated with different sources or receivers, define alternative interpretations of the *semantic objects* in different contexts. Every source or receiver is associated with exactly one context (though not necessarily unique, since different sources or receivers may share the same context). We refer to the collection of context axioms corresponding to a given context c as the *context theory* for c.

The assignment of sources to contexts is modeled explicitly as part of the Context Interchange framework via a source-to-context mapping μ. Thus, $\mu(s) = c$ indicates that the context of source s is given by c. The functional form is chosen over the usual predicate-form (i.e., $\mu(s, c)$) to highlight the fact that every source can only be assigned exactly one context. By abusing the notation slightly, we sometimes write $\mu(r) = c$ if r is a relation in source s. As we shall see later on, the context of *receivers* is modeled explicitly as part of a query.

In the remaining subsections, we describe each of the above components in turn. This is followed by a description of the logical inferences—called abduction—for realizing query mediation. The Context Interchange framework is constructed on a deductive and object-oriented data model (and language) of the family of F(rame) logic [Kifer et al. 1995], which combines both features of object-oriented and deductive data models. The syntax and semantics of this language will be introduced informally throughout the discussion, and we sometimes alternate between an F-logic and a predicate calculus syntax to make the presentation more intuitive. This is no cause for alarm, since it has been repeatedly shown that one syntactic form is equivalent to the other (see, for instance, Abiteboul et al. [1993]). Notwithstanding this, the adoption of an "object-oriented" syntax provides us with greater flexibility in representing and reusing data semantics captured in different contexts. This is instrumental in defining an integration infrastructure that is *scalable*, *extensible*, and *accessible* [Goh et al. 1994]. This

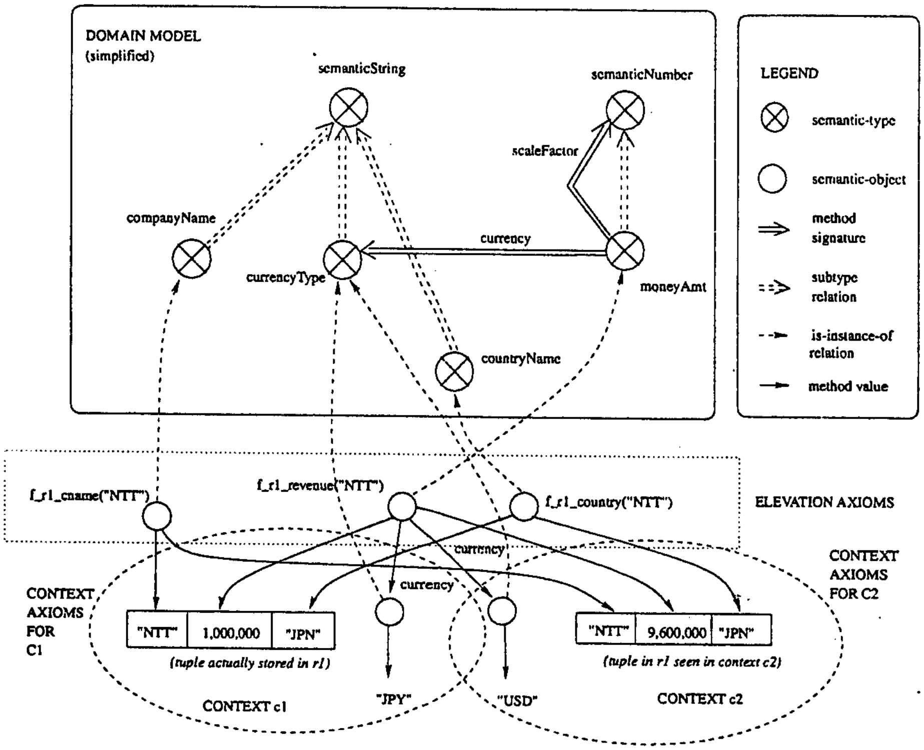


Fig. 3. A graphical illustration of the different components of a Context Interchange framework.

observation will be revisited in Section 4 where we compare our approach to the integration strategy adopted in Carnot [Collet et al. 1991].

3.1 The Domain Model

We distinguish between two kinds of data objects in the COIN data model: *primitive objects*, which are instances of *primitive types*, and *semantic objects* which are instances of *semantic types*. Primitive types correspond to data types (e.g., strings, integers, and reals) which are native to sources and receivers. Semantic types, on the other hand, are complex types introduced to support the underlying integration strategy. Specifically, semantic objects may have properties, called *modifiers*, which serve as annotations that make explicit the semantics of data in different contexts. Every object is identifiable using a unique *object-id* (OID) and has a value (not necessarily distinct). In the case of primitive objects, we do not distinguish between the OID and its value. Semantic objects, on the other hand, may have distinct values in different context. Examples of these will be presented shortly.

A *domain model* is a collection of primitive types and semantic types which provides a common type system for information exchange between disparate systems. A (simplified) domain model corresponding to our motivational example in Section 2 can be seen in Figure 3. We use a different symbol for types and object instances, and different arrow types to illustrate the disparate relationships between these. For example, double-shaft arrows indicate "signatures" and identify what modifiers are defined for each type, as well as the type of the object which can be assigned to the (modifier) slot. The notation used should be self-explanatory from the accompanying legend.

As in other "object-oriented" formalisms, types may be related in an abstraction hierarchy where properties of a type are inherited. This inheritance can be *structural* or *behavioral*: the first refers to the inheritance of the type structure, and the second, that of values assigned to instances of those types. For example, semanticNumber, moneyAmt, and semanticString are all semantic types. Moreover, moneyAmt is a subtype of semanticNumber and has modifiers currency and scaleFactor. If we were to introduce a subtype of moneyAmt, say stockPrice, into this domain model, then stockPrice will inherit the modifiers currency and scaleFactor from moneyAmt by structural inheritance. If we had indicated that all (object) instances of moneyAmt will be reported using a scaleFactor of 1, this would be true of all instances of stockPrice as well by virtue of behavioral inheritance (unless this value assignment is overridden).

The object labeled f_r1_revenue("NTT") is an example of a semantic object, which is an instance of the semantic type moneyAmt (indicated by the dashed arrow linking the two). The token f_r1_revenue("NTT") is the unique OID and is invariant under all circumstances. Semantic objects are "virtual" objects, since they are never physically materialized for query processing, but exist merely for query mediation. As we will demonstrate in the next section, this object is defined by applying a Skolem function on the key-value of a tuple in the source. It is important to point out that a semantic object may have different values in different "contexts." Suppose we introduce two contexts labeled as c1 and c2 which we associate with sources and receiver as indicated in Figure 3. We may write

```
f_r1_revenue("NTT")[value(c1)  →  1000000].
f_r1_revenue("NTT")[value(c2)  →  9600000].
```

The above statements illustrate statements written in the COIN language (COINL), which mirrors closely that of F-logic [Kifer et al. 1995]. The token value(c1) is a *parameterized method* and is said to return the value 1000000 when invoked on the object f_r1_revenue("NTT"). The same statements could have been written using a predicate calculus notation:

```
ist(c1, value(f_r1_revenue("NTT"),1000000)).
ist(c2, value(f_r1_revenue("NTT"),9600000)).
```

The choice of an object logic however allows certain features (e.g., inheritance and overridding) to be represented more conveniently.

3.2 Elevation Axioms

Elevation axioms provide the means for mapping "values" present in sources to "objects" which are meaningful with respect to a domain model.

This is accomplished by identifying the semantic type corresponding to each attribute in the export schema, and in allowing semantic objects to be instantiated from values present in the source. In the graphical interface which is planned for the existing prototype, this is simply accomplished by scrolling through the domain model and "clicking" on the semantic type that corresponds to a given attribute that is to be exported by the current source.

Internally, this mapping of attributes to semantic types is formally represented in two different sets of assertions. We present below the abstract syntax of the language, which emphasizes the "logical" character of our representation. A concrete syntax, a lá OQL, is being developed for end-users and applications programmers to make the representation more accessible.

The first group of axioms introduces a semantic object corresponding to each attribute of a tuple in the source. For example, the statement

$$\forall x\ \forall y\ \forall z\ \exists u \text{ s.t. } u : \texttt{moneyAmt} \leftarrow \texttt{r1}(x, y, z)$$

asserts that there exists some semantic object u of type moneyAmt corresponding to each tuple in relation r1. This statement can be rewritten into the *Horn clause* [Lloyd 1987], where all variables are assumed to be universally quantified:

$$\texttt{f\_r1\_revenue}(x, y, z) : \texttt{moneyAmt} \leftarrow \texttt{r1}(x, y, z).$$

The existentially quantified variable u is replaced by the *Skolem object* [Lloyd 1987] f_r1_revenue(x, y, z). Notice that the *Skolem function* (f_r1_revenue) is chosen such that it is guaranteed to be unique. In this example, it turns out that the functional dependency cname → {revenue, country} holds on r1: this allows us to replace f_r1_revenue(x, y, z) by f_r1_revenue(x) without any loss of generality. This follows trivially from the fact that whenever we have f_r1_revenue(x, y, z) and f_r1_revenue(x, y', z'), it must be that $y = y'$ and $z = z'$ (by virtue of the functional dependency).

The second assertion is needed to provide the assignment of values to the (Skolem) semantic objects created before. We may thus write

$$\texttt{f\_r1\_revenue}(x)[\texttt{value}(c) \rightarrow y] \leftarrow \texttt{r1}(x, y, z),\ \mu(\texttt{r1}, c).$$

Consider, for instance, the semantic object f_r1_revenue("NTT") shown in Figure 3. This object is instantiated via the application of the first assertion. The second assertion allows us to assign the value 1000000 to this object in context c1, which is the context associated with relation r1. The value of this semantic object may however be different in another context, as in the case of c2. The transformation on the values of semantic objects between different contexts is addressed in the next subsection.

3.3 Context Axioms

Context axioms associated with a source or receiver provide for the articulation of the data semantics which are often implicit in the given context.

These axioms come in two parts. The first group of axioms defines the semantics of data at the source or receiver in terms of values assigned to modifiers corresponding to semantic objects. The second group of axioms complements this declarative specification by introducing the "methods" (i.e., *conversion functions*) that define how values of a given semantic object are transformed between different contexts.

Axioms of the first type takes the form of a first-order statement which make assignments to modifiers. Returning to our earlier example, the fact that all moneyAmt in context c2 are reported in US Dollars using a scale-factor of 1 is made explicit in the following axioms:

```
x : moneyAmt, y : semanticNumber ⊢ y[value(c2) → 1]
      ← x [scaleFactor(c2) → y].
x : moneyAmt, y : currencyType ⊢ y[value(c2) → "USD"]
      ← x [currency(c2) → y].
```

In the above statements, the part preceding the symbol " ⊢" constitutes the *predeclaration* identifying the object type(s) (class) for which the axiom is applicable. This is similar to the approach taken in Gulog [Dobbie and Topor 1995]. By making explicit the types to which axioms are attached, we are able to simulate nonmonotonic inheritance through the use of negation, as in Abiteboul et al. [1993].

The semantics of data embedded in a given context may be arbitrarily complex. In the case of context c1, the currency of moneyAmt is determined by the country-of-incorporation of the company which is being reported on. This in turn determines the scale-factor of the amount reported; specifically, money amounts reported using "JPY" uses a scale-factor of 1000, whereas all others are reported in 1's. The corresponding axioms for these are shown below:

```
x :moneyAmt, y :currencyType ⊢ y[value(c1) → v] ←
                 x[currency(c1) → y], x = f_r1_revenue(u),
                 r1(u, _, w), r4(w, v).
x :moneyAmt, y :semanticNumber ⊢ y [value(c1) → 1000] ←
                 x[scaleFactor(c1) → y; currency(c1) → z],
                 z[value(c1) → v], v = "JPY".
x :moneyAmt, y :semanticNumber ⊢ y [value(c1) → 1] ←
                 x[scaleFactor(c1) → y; currency(c1) → z],
                 z[value(c1) → v], v ≠ "JPY".
```

Following Prolog's convention, the token "_" is used to denote an "anonymous" variable. In the first axiom above, r4 is assumed to be in the same context as r1 and is assumed to constitute an ancillary data source for defining part of the context (in this case, the currency used in reporting moneyAmt). Bear in mind also that variables are local to a clause; thus, variables having the same name in different clauses have no relation to one another.

The preceding declarations are not yet sufficient for resolving conflicting interpretations of data present in disparate contexts, since we have yet to define how values of a (semantic) object in one context are to be reported in a different context with different assumptions (i.e., modifier values). This is accomplished in the Context Interchange framework via the introduction of *conversion functions* (methods) which form part of the context axioms. The conversion functions define, for each modifier, how representations of an object of a given type may be transformed to comply with assumptions in the local context. For example, scale-factor conversions in context c1 can be defined by multiplying a given value with the appropriate ratio as shown below:

```
x :moneyAmt ⊢
     x[cvt(scaleFactor,c1)@c,u → v] ←
             x[scaleFactor(c1) →  _[value(c1) → f]],
             x[scaleFactor(c) →  _[value(c1) → g]],
             v = u * g/f.
```

In the "antecedent" of the statement above, the first literal returns the scale-factor of x in context c1. In contrast, the second literal returns the scale-factor of x in some parameterized context c. c and c1 are, respectively, the *source* and *target* context for the tranformation at hand. The objects returned by modifiers (in this case, scaleFactor(c1) and scaleFactor(c)) are semantic objects and need to be dereferenced to the current context before they can be operated upon: this is achieved by invoking the method value(c1) on them. Notice that the same conversion function can be introduced in context c2; the only change required is the systematic replacement of all references to c1 by c2.

The conversion functions defined for semantic objects are invoked when the semantic objects are exchanged between different contexts. For example, the value of the semantic object f_r1_revenue("NTT") in context c2 is given by

```
f_r1_revenue("NTT")[value(c2) → v] ←
     f_r1_revenue("NTT")[cvt(c2) → v].
```

The method cvt(c2) can in turn be rewritten as a series of invocations on the conversion function defined on each modifier pertaining to the semantic type. Thus, in the case of moneyAmt, we would have

```
f_r1_revenue("NTT")[cvt(c2) → w] ←
     f_r1_revenue("NTT")[value(c1) → u],
     f_r1_revenue("NTT")[cvt(currency,c2)@c1,u → v],
     f_r1_revenue("NTT")[cvt(scaleFactor,c2)@c1,v → w].
```

Hence, if the conversion function for currency returns the value 9600, this will be rewritten to 9600000 by the scale-factor conversion function and returned as the value of the semantic object f_r1_revenue("NTT") in context c2.

In the same way whereby r4 is used in the assignment of values to modifiers, ancillary data sources may be used for defining appropriate conversion functions. For instance, currency conversion in context c2 is supported by the relation r3, which provides the exchange rate between two different currencies. In general, the use of ancillary data sources in context axioms will lead to the introduction of additional table lookups in the mediated query, as we have shown earlier in Section 2.

3.4 Query Mediation as Abductive Inferences

The goal of the Context Interchange framework is to provide a formal, logical basis that allows for the automatic mediation of queries such as those described in Section 2. The logical inferences which we have adopted for this purpose can be characterized as *abduction* [Kakas et al. 1993]: in the simplest case, this takes the form

> From observing A and the axiom $B \rightarrow A$
> Infer B as a possible "explanation" of A.

Abductive logic programming (ALP) [Kakas et al. 1993] is an extension of *logic programming* [Lloyd 1987] to support abductive reasoning. Specifically, an *abductive framework* [Eshghi and Kowalski 1989] is a triple $\langle \mathcal{T}, \mathcal{A}, \mathcal{I} \rangle$ where $\mathcal{T}$ is a theory, $\mathcal{I}$ is a set of integrity constraints, and $\mathcal{A}$ is a set of predicate symbols, called *abducible* predicates. Given an abductive framework $\langle \mathcal{T}, \mathcal{A}, \mathcal{I} \rangle$ and a sentence $\exists \vec{X} q(\vec{X})$ (the *observation*), the *abductive task* can be characterized as the problem of finding a *substitution* θ and a set of abducibles Δ, called the *abductive explanation* for the given observation, such that

(1) $\mathcal{T} \cup \Delta \models \forall(q(\vec{X})\theta)$,

(2) $\mathcal{T} \cup \Delta$ satisfies $\mathcal{I}$, and

(3) Δ has some properties that make it "interesting."

Requirement (1) states that Δ, together with $\mathcal{T}$, must be capable of providing an explanation for the observation $\forall(q(\vec{X})\theta)$. The prefix "$\forall$" suggests that all free variables after the substitution are assumed to be universally quantified. The consistency requirement in (2) distinguishes abductive explanations from inductive generalizations. Finally, in the characterization of Δ in (3), "interesting" means primarily that literals in Δ are atoms formed from abducible predicates: where there is no ambiguity, we refer to these atoms also as *abducibles*. In most instances, we would like Δ to also be minimal or nonredundant.

The Context Interchange framework is mapped to an abductive framework $\langle \mathcal{T}, \mathcal{A}, \mathcal{I} \rangle$ in a straightforward manner. Specifically, the domain model axioms, the elevation axioms, and the context axioms are rewritten to normal Horn clauses where nonmonotonic inheritance is simulated through the use of negation. The procedure and semantics for this transformation have been described in Abiteboul et al. [1993]. The resulting set of clauses, together with a handful of generic axioms, defines the theory $\mathcal{T}$ for the abductive framework. The integrity constraints in $\mathcal{I}$ consist of all the integrity constraints defined on the sources complemented with Clark's

Free Equality Axioms[1] [Clark 1978]. Finally, the set of abducibles $\mathcal{A}$ consists of all extensional predicates (relation names exported by sources) and references to externally stored procedures (referenced by some conversion functions).

As we have noted in Section 2, queries in a Context Interchange system are formulated under the assumption that there are no conflicts between sources and/or the receiver. Given an SQL query, context mediation is bootstrapped by tranforming this user query into an equivalent query in the internal COINL representation. For example, the query Q1 (in Section 2) will be rewritten to the following form:

$$\begin{aligned} \text{Q1}^*\text{:}\ & \leftarrow ans(x, y). \\ & ans(x, y) \leftarrow \texttt{r1}(x, y, \_),\ \texttt{r2}(x, z),\ y > z. \end{aligned}$$

The predicate *ans* is introduced so that only those attributes which are needed are projected as part of the answer. This translation is obviously a trivial exercise, since both COINL and relational query languages are variants of predicate calculus.

The preceding query however continues to make reference to primitive objects and (extensional) relations defined on them. To allow us to reason with the different representations built into semantic objects, we introduce two further artifacts which facilitates the systematic rewriting of a query to a form which the context mediator can work with.

—For every extensional relation $\texttt{r}$, we introduce a corresponding *semantic relation* $\bar{\texttt{r}}$ which is isomorphic to the original relation, with each primitive object in the extensional relation being replaced by its semantic object counterpart. For example, the semantic relation for $\bar{\texttt{r}}\texttt{1}$ is defined via the axiom

$$\bar{\texttt{r}}\texttt{1}(\texttt{f\_r1\_cname}(x), \texttt{f\_r1\_revenue}(x), \texttt{f\_r1\_country}(x)) \leftarrow \texttt{r1}(x, \_, \_).$$

A sample tuple of this semantic relation can be seen in Figure 3.

—To take into account the fact that the same semantic object may have different representations in different contexts, we enlarge the notion of classical "relational" comparison operators and insist that such comparisons are only meaningful when they are performed with respect to a given context. Formally, if $\diamond$ is some element of the set $\{=, \neq, \leq, \geq, <, >, \ldots\}$ and x, y are primitive objects or semantic objects (not necessarily of the same semantic type), then we say that

$$x \overset{c}{\diamond} y \text{ iff } (x\ [\texttt{value}(c) \rightarrow u] \text{ and } y\ [\texttt{value}(c) \rightarrow v] \text{ and } u \diamond v)$$

(In the case where both x and y are primitive objects, semantic comparison degenerates to normal relational operations, since the value of a

[1] These consist of the axioms $X = X$ (reflexivity), $X = Z \leftarrow X = Y \wedge Y = Z$ (transitivity), and inequality axioms of the type $a \neq b$, $b \neq c$ for any two non-Skolem terms which do not unify.

primitive object is given by its OID.) The intuition underlying this fabrication is best grasped through an example: in the case of f_r1_revenue("NTT"), we know that

$$\texttt{f\_r1\_revenue("NTT") [value(c1)} \rightarrow \texttt{1000000]}.$$

Thus, the statement f_r1_revenue("NTT") $\overset{c}{<}$ 5000000 is true if c = c1 but not if c = c2 (since f_r1_revenue("NTT")[value(c2) → 9600000]).

Using the above definitions, the context mediator can rewrite the query Q1* shown earlier to the following:

$$ans(u, v) \leftarrow \bar{\texttt{r}}\texttt{1}(x, y, \_),\ \bar{\texttt{r}}\texttt{2}(w, z),\ x \overset{c2}{=} w,\ y \overset{c2}{\leq} z,\ x\texttt{[value(c2)} \rightarrow u\texttt{]},$$
$$y\texttt{[value(c2)} \rightarrow v\texttt{]}.$$

This is obtained by systematic renaming of each extensional predicate (r) to its semantic counterpart ($\bar{\texttt{r}}$), by replacing all comparisons (including implicit "joins") with semantic comparisons, and making sure that attributes which are to be projected in a query correspond to the values of semantic objects in the context associated with the query.

The abductive answer corresponding to the above query can be obtained via backward chaining, using a procedure not unlike the standard *SLD-resolution* procedure [Eshghi and Kowalski 1989]. We present the intuition of this procedure below by visiting briefly the sequence of reasoning in the example query. A formal description of this procedure can be found in Bressan et al. [1997b].

Starting from the query above and resolving each literal with the theory $\mathcal{T}$ in a depth-first manner, we would have obtained the following:

$$\leftarrow \texttt{r1}(u_0, v_0, \_),\ \bar{\texttt{r}}\texttt{2}(w, z),\ \texttt{f\_r1\_cname}(u_0) \overset{c2}{=} w,\ \texttt{f\_r1\_revenue}(u_0) \overset{c2}{\leq} z,$$
$$\texttt{f\_r1\_cname}(u_0)\ \texttt{[value(c2)} \rightarrow u\texttt{]},\ \texttt{f\_r1\_revenue}(u_0)\ \texttt{[value(c2)} \rightarrow v\texttt{]}.$$

The subgoal r1(u_0, v_0, _) cannot be further evaluated and will be abducted at this point, yielding the following sequence:

$$\leftarrow \bar{\texttt{r}}\texttt{2}(w, z),\ \texttt{f\_r1\_cname}(u_0) \overset{c2}{=} w,\ \texttt{f\_r1\_revenue}(u_0) \overset{c2}{\leq} z,$$
$$\texttt{f\_r1\_cname}(u_0)\ \texttt{[value(c2)} \rightarrow u\texttt{]},\ \texttt{f\_r1\_revenue}(u_0)\ \texttt{[value(c2)} \rightarrow v\texttt{]}.$$
$$\leftarrow \texttt{r2}(u', v'),\ \texttt{f\_r1\_cname}(u_0) \overset{c2}{=} w,\ \texttt{f\_r2\_cname}(u'),$$
$$\texttt{f\_r1\_revenue}(u_0) \overset{c2}{\leq} \texttt{f\_r2\_expenses}(u'),$$
$$\texttt{f\_r1\_cname}(u_0)\ \texttt{[value(c2)} \rightarrow u\texttt{]},\ \texttt{f\_r1\_revenue}(u_0)\ \texttt{[value(c2)} \rightarrow v\texttt{]}.$$

Again, r2(u', v') is abducted to yield

$$\leftarrow \texttt{f\_r1\_cname}(u_0) \overset{c2}{=} \texttt{f\_r2\_cname}(u'),$$
$$\texttt{f\_r1\_revenue}(u_0) \overset{c2}{\leq} \texttt{f\_r2\_expenses}(u'),$$
$$\texttt{f\_r1\_cname}(u_0)\ \texttt{[value(c2)} \rightarrow u\texttt{]},\ \texttt{f\_r1\_revenue}(u_0)\ \texttt{[value(c2)} \rightarrow v\texttt{]}.$$

Since companyName has no modifiers, there is no conversion function defined on instances of companyName, so the value of f_r1_cname(u_0) does not vary across any context. Hence, the subgoal f_r1_cname(u_0) $\overset{c2}{=}$ f_r2_cname(u') can be reduced to just $u_0 = u'$ which unifies the variables u and u', reducing the goal further to

$\leftarrow$ f_r1_revenue(u_0) $\overset{c2}{>}$ f_r2_expenses(u_0),
  f_r1_cname(u_0) [value(c2) $\rightarrow u$], f_r1_revenue(u_0) [value(c2) $\rightarrow v$].

This process goes on until this goal list has been reduced to the empty clause. Upon backtracking, alternative abductive answers can be obtained. In this example, we obtain the following abductive answers in direct correspondance to the mediated query MQ1 shown earlier:

Δ_1 = { r1(u, v, _), r2(u, v'), r4(u,"USD"), $v > v'$}
Δ_2 = { r1(u, v_0, _), r2(u, v'), r4(u,"JPY"), r3("JPY","USD",r),
  $v = v_0 * r * 1000$, $v > v'$}
Δ_3 = { r1(u, v_0, _), r2(u, v'), r4(u,y), $y \neq$ "USD", $y \neq$ "JPY",
  r3(y,"USD",r), $v = v_0 * r$, $v > v'$}

The query-rewriting technique described above may also be understood as a form of *partial evaluation*, in which a high-level specification is transformed into a lower-level program which can be executed more efficiently. In this context, the context mediator plays the role of a meta-interpreter that evaluates part of the query (identifying potential conflicts and methods for their resolution in consultation with the logic theory $\mathcal{T}$), while delaying other parts of the query that involve access to extensional databases and evaluable predicates. This compilation can be performed online or offline, i.e., at the time a query is being submitted, or in the form of precompiled view definitions that are regularly queried by users and other client applications.

4. COMPARISON WITH EXISTING APPROACHES

In an earlier report [Goh et al. 1994], we have made detailed comments on the many features that the Context Interchange approach has over traditional *loose-* and *tight-coupling* approaches. In summary, although tightly coupled systems provide better support for data access to heterogeneous systems (compared to loosely coupled systems), they do not scale-up effectively given the complexity involved in constructing a shared schema for a large number of systems and are generally unresponsive to changes for the same reason. Loosely coupled systems, on the other hand, require little central administration but are equally nonviable, since they require users to have intimate knowledge of the data sources being accessed; this assumption is generally nontenable when the number of systems involved is large and when changes are frequent.[2] The Context Interchange approach provides a novel middle ground between the two: it allows knowledge of data semantics to be independently captured in sources and receivers (in the form of context theories), while allowing a specialized

[2] We have drawn a sharp distinction between the two here to provide a contrast of their relative features. In practice, one is most likely to encounter a hybrid of the two strategies. It should however be noted that the two strategies are incongruent in their outlook and are *not* able to easily take advantage of each other's resources. For instance, data semantics encapsulated in a shared schema cannot be easily extracted by a user to assist in formulating a query which seeks to reference the source schemas directly.

mediator (the Context Mediator) to undertake the role of detecting and reconciling potential conflicts at the time a query is submitted.

At a cursory level, the Context Interchange approach may appear similar to many contemporary integration approaches. However, we posit that the similarities are superficial, and that our approach represents a significant departure from these strategies. Given the proliferation of system prototypes, it is not practical to compare our approach with each of these. The following is a sampling of contemporary systems which are representative of various alternative integration approaches.

A number of contemporary systems (e.g., Pegasus [Ahmed et al. 1991], the ECRC Multidatabase Project [Jonker and Schütz 1995], SIMS [Arens and Knoblock 1992], and DISCO [Tomasic et al. 1995]) have attempted to rejuvenate the loose- or tight-coupling approach through the adoption of an object-oriented formalism. For loosely coupled systems, this has led to more expressive data transformation (e.g., O*SQL [Litwin 1992]); in the case of tightly coupled systems, this helps to mitigate the effects of complexity in schema creation and change management through the use of abstraction and encapsulation mechanisms. Although the Context Interchange strategy embraces "object orientation" for the same reasons, it differs by not requiring pairwise reconciliation of semantic conflicts to be incorporated as part of the shared schema. For instance, our approach does not require the domain model to be updated each time a new source is added; this is unlike tightly coupled systems where the shared schema needs to be updated by-hand each time such an event occurs, even when conflicts introduced by the new source are identical to those which are already present in existing sources. Yet another difference is that although a deductive object-oriented formalism is also used in the Context Interchange approach, "semantic objects" in our case exist only conceptually and are never actually materialized during query evaluation. Thus, unlike some other systems (e.g., the ECRC prototype), we do not require an intermediary "object store" where objects are instantiated before they can be processed. In our implementation, both user queries and their mediated counterpart are relational. The mediated query can therefore be executed by a classical relational DBMS without the need to reinvent a query-processing subsystem.

In the Carnot system [Collet et al. 1991], semantic interoperability is accomplished by writing *articulation axioms* which translate "statements" which are true in individual sources to statements which are meaningful in the Cyc knowledge base [Lenat and Guha 1989]. A similar approach is adopted in Faquhar et al. [1995], where it is suggested that domain-specific *ontologies* [Gruber 1991], which may provide additional leverage by allowing the ontologies to be shared and reused, can be used in place of Cyc. While we like the explicit treatment of contexts in these efforts and share their concern for sustaining an infrastructure for data integration, our realization of these differs in several important ways. First, our domain model is a much more impoverished collection of rich types compared to the richness of the Cyc knowledge base. Simplicity is a feature here because the construction of a rich and complex shared model is laborious and error prone, not to mention that it is almost impossible to maintain. Second, the translation of sentences from one context to another is embedded in axioms present in individual context theories, and are not part of the domain model. This means that there is greater scope for different users to introduce conversion functions which are most appropriate for their purposes without requiring these differences to be accounted for globally. Finally, semantics of data is represented in an "object-centric" manner as opposed to a "sentential" representation. For example, to relate two statements (σ and σ') in different distinct contexts c and c', a lifting axiom of the form

$$ist(c, \sigma) \Leftrightarrow ist(c', \sigma')$$

will have to be introduced in Cyc. In the Context Interchange approach, we have opted for a "type-based" representation where conversion functions are attached to types in different contexts. This mechanism allows for greater sharing and reuse of semantic encoding. For example, the same type may appear many times in different predicates (e.g., consider the type `moneyAmt` in a financial application). Rather than writing a lifting axiom for each predicate that redundantly describes how different reporting currencies are resolved, we can simply associate the conversion function with the type `moneyAmt`.

Finally, we remark that the TSIMMIS [Papakonstantinou et al. 1995; Quass et al. 1995] approach stems from the premise that information integration could not, and should not, be fully automated. With this in mind, TSIMMIS opted in favor of providing both a framework and a collection of tools to assist humans in their information processing and integration activities. This motivated the invention of a "lightweight" object model which is intended to be *self-describing*. For practical purposes, this translates to the strategy of making sure that attribute labels are as descriptive as possible and opting for free-text descriptions ("man-pages") which provide elaborations on the semantics of information encapsulated in each object. We concur that this approach may be effective when the data sources are ill structured and when consensus on a shared vocabulary cannot be achieved. However, there are also many other situations (e.g., where data sources are relatively well structured and where *some* consensus can be reached) where human intervention is not appropriate or necessary: this distinction is primarily responsible for the different approaches taken in TSIMMIS and our strategy.

5. CONCLUSION

Although there had been previous attempts at formalizing the Context Interchange strategy (see, for instance, Sciore et al. [1994]), a tight integration of the representational and reasoning formalisms has been consistently lacking. This article has filled this gap by introducing a well-founded logical framework for capturing context knowledge and in demonstrating that query mediation can be formally understood with reference to current work in abductive logic programming. The advancements made in this theoretical frontier have been instrumental in the development of a prototype which provides for the integration of data from disparate sources accessible on the Internet. The architecture and features of this prototype have been reported in Bressan et al. [1997a] and will not be repeated here due to space constraints.

The adoption of a declarative encoding of data semantics brings about other side benefits, chief among which is the ability to query directly the semantics of data which are implicit in different systems. Consider, for instance, the query formulated on the motivational example introduced earlier in the article, that is based on a superset of SQL:[3]

```
Q2: SELECT r1.cname, r1.revenue.scaleFactor IN c1,
           r1.revenue.scaleFactor IN c2 FROM r1
    WHERE r1.revenue.scaleFactor IN c1
          ⟨⟩ r1.revenue.scaleFactor IN c2;
```

Intuitively, this query asks for companies for which scale-factors for reporting "revenue" in r1 (in context c1) differ from that which the user assumes (in context c2). We refer to queries such as Q2 as *knowledge-level queries*, as opposed to *data-level queries* which are enquires on factual data present in data sources. Knowledge-level queries have received little attention in the database literature and to our knowledge have not been addressed by the data integration community. This is a significant gap in the literature given that heterogeneity in disparate data sources arises primarily from incompatible assumptions about how data are interpreted. Our ability to integrate access to both data and semantics can be exploited by users to gain insights into differences among particular systems; for example, we may want to know "Do sources A and B report a piece of data differently? If so, how?" Alternatively, this facility may be exploited by a query optimizer which may want to identify sites with minimal conflicting interpretations in identifying a query plan which requires less costly data transformations.

Interestingly, knowledge-level queries can be answered using the exact same inference mechanism for mediating data-level queries. Hence, submitting query Q2 to the Context Mediator will yield the result

```
MQ2: SELECT r1.cname, 1000, 1 FROM r1, r4
     WHERE r1.country = r4.country AND r4.currency = 'JPY';
```

which indicates that the answer consists of companies for which the reporting currency attribute is 'JPY', in which case the scale-factors in context c1 and c2 are 1000 and 1 respectively. If desired, the mediated query MQ2 can be evaluated on the extensional data set to return an answer grounded in the extensional data set. Hence, if MQ2 is evaluated on

[3] Sciore et al. [1992] have described a similar (but not identical) extension of SQL in which context is treated as a "first-class object." We are not concerned with the exact syntax of such a language here; the issue at hand is how we might support the underlying inferences needed to answer such queries.

the data set shown in Figure 1, we would obtain the singleton answer ⟨'NTT', 1000, 1⟩.

Yet another feature of Context Interchange is that *answers* to queries can be both intensional and extensional. Extensional answers correspond to fact sets which one normally expects of a database retrieval. Intensional answers, on the other hand, provide only a characterization of the extensional answers *without* actually retrieving data from the data sources. In the preceding example, MQ2 can in fact be understood as an intensional answer for Q2, while the tuple obtained by the evaluation of MQ2 constitutes the extensional answer for Q2.

As seen from the above example, intensional answers are grounded in extensional predicates (i.e., names of relations), evaluable predicates (e.g., arithmetic operators or "relational" operators), and external functions which can be directly evaluated through system calls. The intensional answer is thus no different from a query which can normally be evaluated on a conventional query subsystem of a DBMS. Query answering in a Context Interchange system is thus a two-step process: an intensional answer is first returned in response to a user query; this can then be executed on a conventional query subsystem to obtain the extensional answer.

The intermediary intensional answer serves a number of purposes [Imielinski 1987]. Conceptually, it constitutes the mediated query corresponding to a user query and can be used to confirm the user's understanding of what the query actually entails. More often than not, the intensional answer can be more informative and easier to comprehend compared to the extensional answer it derives. (For example, the intensional answer MQ2 actually conveys more information than merely the extensional answer comprising a single tuple.) From an operational standpoint, the computation of extensional answers is likely to be many orders of magnitude more expensive compared to the evaluation of the corresponding intensional answer. It therefore makes good sense not to continue with query evaluation if the intensional answer satisfies the user. From a practical standpoint, this two-stage process allows us to separate query mediation from query optimization and execution. As we have illustrated in this article, query mediation is driven by logical inferences which do not bond well with (predominantly cost-based) optimization techniques that have been developed [Mumick and Pirahesh 1994; Seshadri et al. 1996]. The advantage of keeping the two tasks apart is thus not merely a conceptual convenience, but allows us to take advantage of mature techniques for query optimization in determining how best a query can be evaluated.

To the best of our knowledge, the application of abductive reasoning to "database problems" has been confined to the view-update problem [Kakas and Mancarella 1990]. Our use of abduction for query rewriting represents a potentially interesting avenue which warrants further investigation. For example, consistency checking performed in the abduction procedure allows a mediated query to be pruned to arrive at intensional answers which are more comprehensible as well as queries which are more efficient. This bears some similarity to techniques developed for *semantic query optimization* [Chakravarthy et al. 1990] and appears to be useful for certain types of optimization problems.

ACKNOWLEDGMENTS

We like to thank Umesh Dayal, Michael Kifer, Arnie Rosenthal, Edward Sciore, and the anonymous referees for their feedback on earlier drafts of this article.

In Memory of Cheng Hian Goh (1965–1999)

Cheng Hian Goh passed away on April 1, 1999, at the age of 33. He is survived by his wife Soh Mui Lee and two sons, Emmanuel and Gabriel, to whom we all send our deepest condolences.

Cheng Hian received his PhD in Information Technologies from the Massachusetts Institute of Technology in February, 1997. He joined the Department of Computer Science, National University of Singapore as an Assistant Professor in November, 1996.

He loved and was dedicated to his works—teaching as well as research. He believed in giving his students the best and what they deserved. As a young database researcher, he had made major contributions to the field as testified by his publications (in ICDE'97, VLDB'98, and ICDE'99).

Cheng Hian was a very sociable person, and often sought the company of friends. As such, he was loved by those who come in contact with him. He would often go the extra mile to help his friends and colleagues. He was a great person, and had touched the lives of many. We suddenly realized that there are many things that we will never do together again. We will miss him sorely, and his laughter, and his smile...

REFERENCES

Abiteboul, S., Lausen, G., Uphoff, H., and Waller, E. 1993. Methods and rules. *SIGMOD Rec. 22*, 2 (June 1, 1993), 32–41.

Ahmed, R., De Smedt, P., Du, W., Kent, W., Ketabchi, M. A., Litwin, W. A., Rafii, A., and Shan, M.-C. 1991. The Pegasus heterogeneous multidatabase system. *IEEE Comput. 24*, 12 (Dec. 1991), 19–27.

Arens, Y. and Knoblock, C. A. 1992. Planning and reformulating queries for semantically-modeled multidatabase systems. In *Proceedings of the 1st International Conference on Information and Knowledge Management* (CIKM-92, Baltimore, MD, Nov.), Y. Yesha, Ed. 92–101.

Bressan, S., Goh, C. H., Fynn, K., Jakobisiak, M., Hussein, K., Kon, H., Lee, T., Madnick, S., Pena, T., Qu, J., Shum, A., and Siegel, M. 1997a. The Context Interchange mediator prototype. *SIGMOD Rec. 26*, 2, 525–527.

Bressan, S., Goh, C. H., Lee, T., Madnick, S., and Siegel, M. 1997b. A procedure for mediation of queries to sources in disparate contexts. In *Proceedings of the 1997 international symposium on Logic programming* (ILPS '97, Port Washington, NY, Oct. 12–16, 1997), J. Małuszyński, I. V. Ramakrishnan, and T. Swift, Eds. MIT Press, Cambridge, MA, 213–227.

Catarci, T. and Lenzerini, M. 1993. Representing and using interschema knowledge in cooperative information systems. *Int. J. Intell. Coop. Inf. Syst. 2*, 4 (Dec.), 375–399.

Chakravarthy, U. S., Grant, J., and Minker, J. 1990. Logic-based approach to semantic query optimization. *ACM Trans. Database Syst. 15*, 2 (June 1990), 162–207.

CLARK, K. 1978. Negation as failure. In *Logic and Data Bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York, NY, 292–322.

COLLET, C., HUHNS, M. N., AND SHEN, W.-M. 1991. Resource integration using a large knowledge base in Carnot. *IEEE Comput. 24*, 12 (Dec. 1991), 55–62.

DOBBIE, G. AND TOPOR, R. 1995. On the declarative and procedural semantics of deductive object-oriented systems. *J. Intell. Inf. Syst. 4*, 2 (Mar. 1995), 193–219.

ESHGHI, K. AND KOWLASKI, R. A. 1989. Abduction compared with negation by failure. In *Proceedings of the 6th International Conference on Logic Programming* (Lisbon, Spain). 234–255.

FAQUHAR, A., DAPPERT, A., FILKES, R., AND PRATT, W. 1995. Integrating information sources using context logic. In *Proceedings of the AAAI-95 Spring Symposium on Information Gathering from Distributed Heterogeneous Environments*. AAAI Press, Menlo Park, CA.

GOH, C. H., BRESSAN, S., MADNICK, S. E., AND SIEGAL, M. D. 1996. Context interchange: Representing and reasoning about data semantics in heterogeneous systems. Sloan School Working Paper No. 3928. MIT-Alfred P. Sloan School of Management, Cambridge, MA.

GOH, C. H., MADNICK, S. E., AND SIEGEL, M. D. 1994. Context interchange: Overcoming the challenges of large-scale interoperable database systems in a dynamic environment. In *Proceedings of the 3rd International Conference on Information and Knowledge Management* (CIKM '94, Gaithersburg, Maryland, Nov. 29–Dec. 2, 1994), N. R. Adam, B. K. Bhargava, and Y. Yesha, Eds. ACM Press, New York, NY, 337–346.

GRUBER, T. R. 1991. The role of common ontology in achieving sharable, reusable knowledge bases. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference* (Cambridge, MA.), J. A. Allen, R. Files, and E. Sandewall, Eds. Morgan Kaufmann, San Mateo, California, 601–602.

HULL, R. 1997. Managing semantic heterogeneity in databases: a theoretical prospective. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems* (PODS '97, Tucson, Arizona, May 12–14, 1997), A. Mendelzon and Z. M. Özsoyoglu, Eds. ACM Press, New York, NY, 51–61.

IMEILINSKI, T. 1987. Intelligent query answering in rule based systems. *J. Logic Program. 4*, 3 (Sept. 1987), 229–257.

JONKER, W. AND SCHÜTZ, H. 1995. The ECRC multi database system. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* (SIGMOD '95, San Jose, CA, May 23–25, 1995), M. Carey and D. Schneider, Eds. ACM Press, New York, NY, 490.

KAKAS, A. C. AND MANCARELLA, P. 1990. Database updates through abduction. In *Proceedings of the 16th International Conference on Very Large Databases* (Brisbane, Australia, Aug. 13–16, 1990), D. McLeod, R. Sacks-Davis, and H. Schek, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 650–661.

KAKAS, A. C., KOWALSKI, R. A., AND TONI, F. 1993. Abductive logic programming. *J. Logic Program. 2*, 6, 719–770.

KIFER, M., LAUSEN, G., AND WU, J. 1995. Logical foundations of object-oriented and frame-based languages. *J. ACM 42*, 4 (July 1995), 741–843.

KUHN, E. AND LUDWIG, T. 1988. VIP-MDBS: A logic multidatabase system. In *International Symposium on Databases in Parallel and Distributed Systems* (Austin, Texas, Dec. 5-7, 1988), J. E. Urban, Ed. IEEE Computer Society Press, Los Alamitos, CA, 190–201.

LANDERS, T. AND ROSENBERG, R. 1982. An overview of Multibase. In *Proceedings of the 2nd International Symposium for Distributed Databases*. 153–183.

LENAT, D. B. AND GUHA, R. V. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Addison-Wesley Publishing Co., Inc., Redwood City, CA.

LEVY, A. Y., MENDELZON, A. O., AND SAGIV, Y. 1995a. Answering queries using views (extended abstract). In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS '95, San Jose, California, May 22–25, 1995), M. Yannakakis, Ed. ACM Press, New York, NY, 95–104.

LEVY, A. Y., SRIVASTAVA, D., AND KIRK, T. 1995b. Data model and query evaluation in global information systems. *J. Intell. Inf. Syst. 5*, 2 (Sept. 1995), 121–143.

LITWIN, W. 1992. O*SQL: A language for object oriented multidatabase interoperability. In *Proceedings of the Conference on IFIP WG2.6 Database Semantics and Interoperable Database Systems (DS-5)* (Lorne, Victoria, Australia), D. K. Hsiao, E. J. Neuhold, and R. Sacks-Davis, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands, 119–138.

LITWIN, W. AND ABDELLATIF, A. 1987. An overview of the multi-database manipulation language MDSL. *Proc. IEEE 75*, 5, 621–632.

LLOYD, J. W. 1987. *Foundations of Logic Programming*. 2nd ed. Springer-Verlag Symbolic Computation and Artificial Intelligence Series. Springer-Verlag, Vienna, Austria.

MCCARTHY, J. 1987. Generality in artificial intelligence. *Commun. ACM 30*, 12 (Dec. 1987), 1030–1035.

MUMICK, I. S. AND PIRAHESH, H. 1994. Implementation of magic-sets in a relational database system. *SIGMOD Rec. 23*, 2 (June 1994), 103–114.

PAPAKONSTANTINOU, Y., GARCIA-MOLINA, H., AND WIDOM, J. 1995. Object exchange across heterogeneous information sources. In *Proceedings of the IEEE International Conference on Data Engineering* (Mar.). IEEE Press, Piscataway, NJ.

QUASS, D., RAJARAMAN, A., SAGIV, Y., ULLMAN, J., AND WIDON, J. 1995. Querying semistructured heterogeneous information. In *Proceedings of the 4th International Conference on Deductive and Object-Oriented Databases* (Singapore, Dec.). Springer-Verlag, Berlin, Germany.

SCIORE, E., SIEGAL, M., AND ROSENTHAL, A. 1992. Context interchange using meta-attributes. In *Proceedings of the 1st International Conference on Information and Knowledge Management* (CIKM-92, Baltimore, MD, Nov.), Y. Yesha, Ed. 377–386.

SCIORE, E., SIEGEL, M., AND ROSENTHAL, A. 1994. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Trans. Database Syst. 19*, 2 (June 1994), 254–290.

SESHADRI, P., HELLERSTEIN, J. M., PIRAHESH, H., LEUNG, T. C., RAMAKRISHNAN, R., SRIVASTAVA, D., STUCKEY, P. J., AND SUDARSHAN, S. 1996. Cost-based optimization for magic: Algebra and implementation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (SIGMOD '96, Montreal, Canada). ACM, New York, NY, 435–446.

SHETH, A. P. AND LARSON, J. A. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv. 22*, 3 (Sept. 1990), 183–236.

SIEGEL, M. AND MADNICK, S. E. 1991. A metadata approach to resolving semantic conflicts. In *Proceedings of the 17th Conference on Very Large Data Bases* (Barcelona, Spain, Sept.). VLDB Endowment, Berkeley, CA, 133–145.

TEMPLETON, M., BRILL, D., DAO, S. K., LUND, E., WARD, P., CHEN, A. L. P., AND MACGREGOR, R. 1987. Mermaid—a front end to distributed heterogeneous databases. *Proc. IEEE 75*, 5, 695–708.

TOMASIC, A., RASCHID, L., AND VALDURIEZ, P. 1996. Scaling heterogeneous databases and the design of DISCO. In *Proceedings of the 16th IEEE International Conference on Distributed Computing Systems* (Hong Kong, May). IEEE Computer Society Press, Los Alamitos, CA.

ULLMAN, J. D. 1997. Information integration using logical views. In *Proceedings of the 6th International Conference on Database Theory* (ICDT '97, Delphi, Greece, Jan.). Springer-Verlag, Berlin, Germany, 19–40.

WIEDERHOLD, G. 1992. Mediators in the architecture of future information systems. *IEEE Comput. 25*, 3 (Mar. 1992), 38–49.

Received: June 1997; revised: April 1998 and June 1998; accepted: August 1998

3. ARCHITECTURE AND IMPLEMENTATION

Context Knowledge Representation and Reasoning in the Context Interchange System*

Stephane Bressan[1], Cheng Goh[2], Natalia Levina, Stuart Madnick, Ahmed Shah, Michael Siegel

Massachusetts Institute of Technology, Cambridge, MA 02139

May 21, 1999 – SM revisions

Abstract

The Context Interchange Project presents a unique approach to the problem of semantic conflict resolution among multiple heterogeneous data sources. The system presents a semantically meaningful view of the data to the receivers (e.g. user applications) for all the available data sources. The semantic conflicts are automatically detected and reconciled by a Context Mediator using the context knowledge associated with both the data sources and the data receivers. The results are collated and presented in the receiver context. The current implementation of the system provides access to flat files, classical relational databases, on-line databases, and web services. An example application, using actual financial information sources, is described along with a detailed description of the operation of the system for an example query.

1. Introduction

In recent years the amount of information available has grown exponentially. While the availability of so much information has helped people become self-sufficient and get access to all the information handily, this has created another dilemma. All these data sources and the technologies that are employed by the data source providers do not provide sufficient logical connectivity (the ability to exchange data meaningfully). Logical connectivity is crucial because users of these sources expect each system to understand requests stated in their own terms, using their own concepts of how the world is defined and structured. As a result, any data integration effort must be capable of reconciling semantic conflicts among sources and receivers. This problem is generally referred to as the need for *semantic interoperability* among distributed data sources.

The Context Interchange Project at MIT [1,2] is studying the semantic integration of disparate information sources. Like other information integration projects (the SIMS project at ISI [3], the TSIMMIS project at Stanford [4], the DISCO project at Bull-INRIA [5], the Information Manifold project at At&T [6], the Garlic project at IBM [7], the Infomaster project at Stanford [8]), we have adopted a Mediation architecture as outlined in Wiederhold's seminal paper [9].

In section 2, we present a motivational scenario of a user trying to access information from various actual data sources and the problems faced. Section 3 describes the current implementation of the Context mediation system. Section 4 presents a detailed discussion of the various subsystems, highlighting the context knowledge representation and reasoning, using the scenario outlined in section 2. Section 5 concludes our discussion.

2. Why Context Mediation ? – An Example Scenario

Consider an example of a financial analyst doing research on Daimler Benz. She needs to find out the net income, net sales, and total assets of Daimler Benz Corporation for the year ending 1993. In addition to that, she needs to know the closing stock price of Daimler Benz. She normally uses the financial data stored in the *Worldscope*[3] database. She recalls Jill, her co-worker telling her about two other databases, *Datastream*[4] and *Disclosure*[5] and how they contained much of the information that Jill needed. She starts off with *Worldscope* database. She knows that *Worldscope* has total assets for all the companies. She brings up a query tool and issues a query:

* This work is supported in part by DARPA and USAF/Rome Laboratory under contract F30602-93-C-0160.

[1] Now at the National University of Singapore.

[2] Now at the National University of Singapore.

[3] The *Worldscope* database is an extract from the Worldscope financial data source

[4] The *Datastream* database is an extract from the Datastream financial data source.

[5] The *Disclosure* database, once again, is an extract from the original Disclosure financial data source. By coincidence, although all three sources were originally provided by independent companies, they are all currently owned by a single company, Primark.

```
select company_name, total_assets from worldscope
where company_name = "DAIMLER-BENZ AG";
```

She immediately gets back the result:

DAIMLER-BENZ AG 5659478

Satisfied, she moves on and figures out after looking at the data information for the new databases that she can get the data on net income from *Disclosure* and net sales from *Datastream*. For net income, she issues the query:

```
select company_name, net_income from disclosure
where company_name = "DAIMLER-BENZ AG";
```

The query does not return any records. Puzzled, she checks for typos and tries again. She knows that the information exists. She tries one more time, this time entering a partial name for DAIMLER BENZ.

```
select company_name, net_income from disclosure
where company_name like "DAIMLER%";
```

She gets the record back:

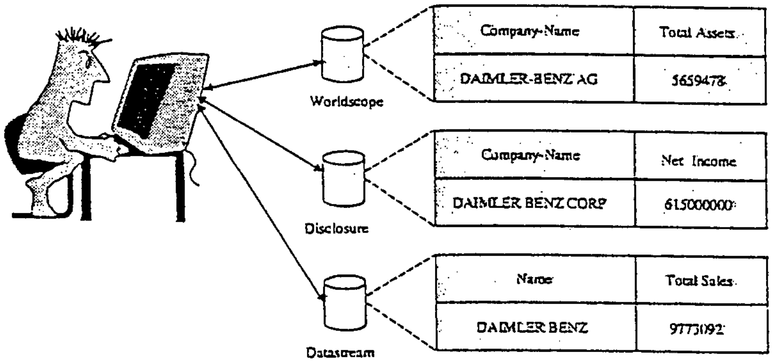
DAIMLER BENZ CORP   615000000

She now realizes that the data sources do not conform to the same standards, as it becomes obvious from the names. Cautious, she presses on and issues the third query:

```
select name, total_sales from datastream
where  name like "DAIMLER%";
```

She gets the result:

DAIMLER-BENZ   9773092



Company-Name	Total Assets
DAIMLER-BENZ AG	5659478

Company-Name	Net Income
DAIMLER BENZ CORP	615000000

Name	Total Sales
DAIMLER BENZ	9773092

Figure 1

As she is putting the results together, she realizes that there are a number of things unusual about the data set shown in Figure 1. First of all, the Total Sales are twice as much as the total assets of the company, which is highly unlikely for a company like Daimler Benz. What is even more disturbing is that net income is more than 60 times as much as total sales. She immediately realizes something is wrong and grudgingly opens up the documents that came with the databases. Upon studying the documentation, she finds out some interesting facts about the data that she was using so gaily. She finds out that *Datastream* has a scale factor of 1000 for all the financial amounts, while *Disclosure* uses a scale factor of 1. In addition, both *Disclosure* and *Datastream* use the country of incorporation to identify the currency, which, in the case of Daimler-Benz, would be German Deutschmarks. She knew that *Worldscope* used a scale factor of 1000 but at least everything was in U.S Dollars. Now she has to reconcile all the information by finding a data source (possibly on the web) that contains the historical currency exchange rates (i.e. as of end of the year 1993). In addition she still has to somehow find another data source to get the latest stock price for Daimler Benz. For that, she knows she will first have to find out the ticker for Daimler Benz and then look up the price using one of the many stock quote servers on the web.

The Context Mediation system can be used to automatically detect and resolve all the semantic conflicts between all the data sources being used and can present the results to the user in the format that she is familiar with. In the above example, if the analyst were using the Context Mediation system instead, all she had to do was formulate and ask a single query without having to worry about the underlying differences between the data. Both her request and the result would be formulated in her preferred context (e.g. *Worldscope*). The multi-source query, **Query1**, could be stated as follows:

```
select worldscope.total_assets; datastream.total_sales,
disclosure.net_income, quotes.Last
from worldscope, datastream, disclosure, quotes  where
worldscope.company_name = "DAIMLER-BENZ AG" and
datastream.as_of_date = "01/05/94"  and
worldscope.company_name = datastream.name and
worldscope.company_name = disclosure.company_name and
worldscope.company_name = quotes.cname ;
```

The system would then detect and reconcile the conflicts encountered by the analyst.

3. Overview of the COIN Project

The COntext INterchange (COIN) strategy seeks to address the problem of *semantic interoperability* by consolidating distributed data sources and providing a unified view. COIN technology presents all data sources as SQL relational databases by providing generic wrappers for them. The underlying integration strategy, called the COIN model, defines a novel approach for mediated [9] data access in which semantic conflicts among heterogeneous systems are automatically detected and reconciled by the *Context Mediator*.

3.1 The COIN Framework

The COIN framework is composed of both a data model and a logical language, COINL [11], derived from the family of F-Logic [10]. The data model and language are used to define the *domain model* of the receiver and data source and the *context* [12] associated with them. The data model contains the definitions for the "types" of information units (called *semantic types*) that constitute a common vocabulary for capturing the semantics of data in disparate systems. *Contexts*, associated with both information sources and receivers, are collections of statements defining how data should be interpreted and how potential conflicts (differences in the interpretation) should be resolved. Concepts such as *semantic-objects, attributes, modifiers*, and *conversion functions* define the semantics of data inside and across *contexts*. Together with the deductive and object-oriented features inherited from F-Logic, the COIN data model and COINL constitute an appropriate and expressive framework for representing semantic knowledge and reasoning about semantic heterogeneity.

3.2 Context Mediator

The *Context Mediator* is the heart of the COIN project. Mediation is the process of rewriting queries posed in the receiver's *context* into a set of mediated queries where all actual conflicts are explicitly resolved and the result is reformulated in the receiver context. This process is based in an abduction [13] procedure that determines what information is needed to answer the query and how conflicts should be resolved by using the axioms in the different *contexts* involved. Answers generated by the mediation unit can be both extensional and intentional. Extensional answers correspond to the actual data retrieved from the various sources involved. Intentional answers, on the other hand, provide only a characterization of the extensional answer without actually retrieving data from the data sources. In addition, the mediation process supports queries on the semantics of data that are implicit in the different systems. There are referred to as *knowledge-level queries* as opposed to *data-level queries* that are enquires on the factual data present in the data sources. Finally, integrity knowledge on one source or across sources can be naturally involved in the mediation process to improve the quality and information content of the mediated queries and ultimately aid in the optimization of the data access.

3.3 System Perspective

From a system perspective, the COIN strategy combines the best features of the *loose-* and *tight-coupling* approaches to *semantic interoperability* [14] among autonomous and heterogeneous systems. Its modular design and implementation, depicted in Figure 2, funnels the complexity of the system into manageable chunks, enables sources and receivers to remain loosely-coupled to one another, and sustains an infrastructure for data integration.
This modularity, both in the components and the protocol, also keeps our infrastructure scalable, extensible, and accessible [2]. By *scalability*, we mean that the complexity of creating and administering the mediation services does not increase exponentially with the number of sources and receivers that participate. *Extensibility* refers to the

ability to incorporate changes into the system in a graceful manner; in particular, local changes do not have adverse effects on other parts of the system. Finally, *accessibility* refers to how a user, in terms of its ease-of-use, perceives the system and flexibility in supporting a variety of queries.

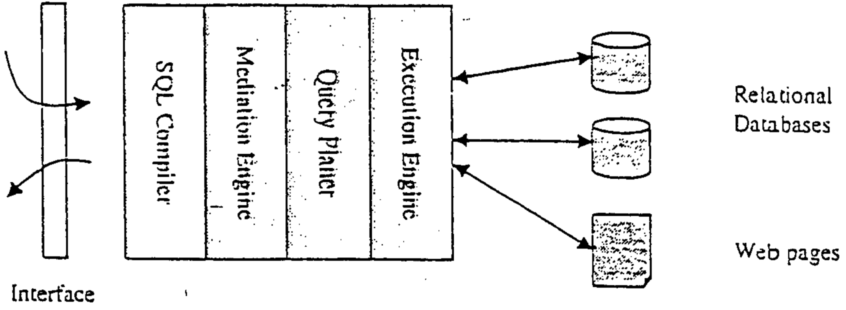


Figure 2: Context Mediator

3.4 Application Domains

The COIN technology can be applied to a variety of scenarios where information needs to be shared amongst heterogeneous sources and receivers. The need for this novel technology in the integration of disparate data sources can be readily seen in many examples.

We have already seen one application of context mediation technology in the financial domain in the previous section. There are many information providers that provide historical data and other research both to institutions (investment banks, brokerages) as well as individual investors. Most of the time this information is presented in different formats and must be interpreted with different rules. Obvious examples are scale-factors and currency of monetary figures. Much more subtle mismatches of assumptions across sources or even inside one source can be critical in the process of financial decision making. Many such examples have been discovered as part of this research effort.

In the domain of manufacturing inventory control, the ability to access design, engineering, manufacturing and inventory data pertaining to all parts, components, and assemblies vital to any large manufacturing process. Typically, thousands of contractors play roles and each contractor tends to set up its data in its own individualistic manner. Managers may need to reconcile inputs received from various contractors in order to optimize inventory levels and ensure overall productivity and effectiveness. As another example, the modern health care enterprise lies at the nexus of several different industries and institutions. Within a single hospital, different departments (e.g. internal medicine, medical records, pharmacy, admitting, and billing) maintain separate information systems yet must share data in order to ensure high levels of care. Medical centers and local clinics not only collaborate with one another but with State and Federal regulators, insurance companies, and other payer institutions. This sharing requires reconciling differences such as those of procedure codes, medical supplies, classification schemes, and patient records. Similar situations have been found in almost every industry. Other industries studied in this research effort include government and military organizations.

4. The COIN Architecture

The feasibility and features of this proposed strategy have been demonstrated in a working system that provides mediated access to both on-line structured databases and semi-structured data sources such as web sites. The infrastructure leverages on the World Wide Web in a number of ways. First, COIN relies on the hypertext transfer protocol for the physical connectivity among sources and receivers and the different mediation components and services. Second, COIN employs the hypertext markup Language and Java for the development of portable user interfaces. Figure 3 shows the architecture of the COIN system. It consists of three distinct groups of processes.

- Client Processes provide the interaction with receivers and route all database requests to the Context Mediator. An example of a client process is the *multi-database browser* [15], which provides a point-and-click interface for formulating queries to multiple sources and for displaying the answers obtained. Specifically, any application program that issues queries to one or more sources can be considered a client process.

- **Server Processes** refer to *database gateways* and *wrappers*. Database gateways provide physical connectivity to a database on a network. The goal is to insulate the Mediator Process from the idiosyncrasies of different database management systems by providing a uniform protocol for database access as well as canonical query language (and data model) for formulating the queries. Wrappers provide richer functionality by allowing semi-structured documents on the World Wide Web to be queried as if they were relational databases. This is accomplished by defining an *export schema* for each of these web sites and describing how attribute-values can be extracted from a web site using a finite automaton with pattern matching [16].
- **Mediator Processes** refer to the system components that collectively provide the mediation services. These include SQL-to-datalog compiler, context mediator, and query planner/optimizer and multi-database executioner. SQL-to-Datalog compiler translates a SQL query into its corresponding datalog format. The Context Mediator rewrites the user-provided query into a mediated query with all the conflicts resolved. The planner/optimizer produces a query evaluation plan based on the mediated query. The multi-database executioner executes the query plan generated by the planner. It dispatches sub-queries to the server processes, collates the intermediary results, converts the result into the client context, and returns the reformulated answer to the client processes.

Of these three distinct groups of processes, the most relevant to our discussion of context knowledge and reasoning are the mediator processes. We will start by explaining the domain model and then discuss the prototype system.

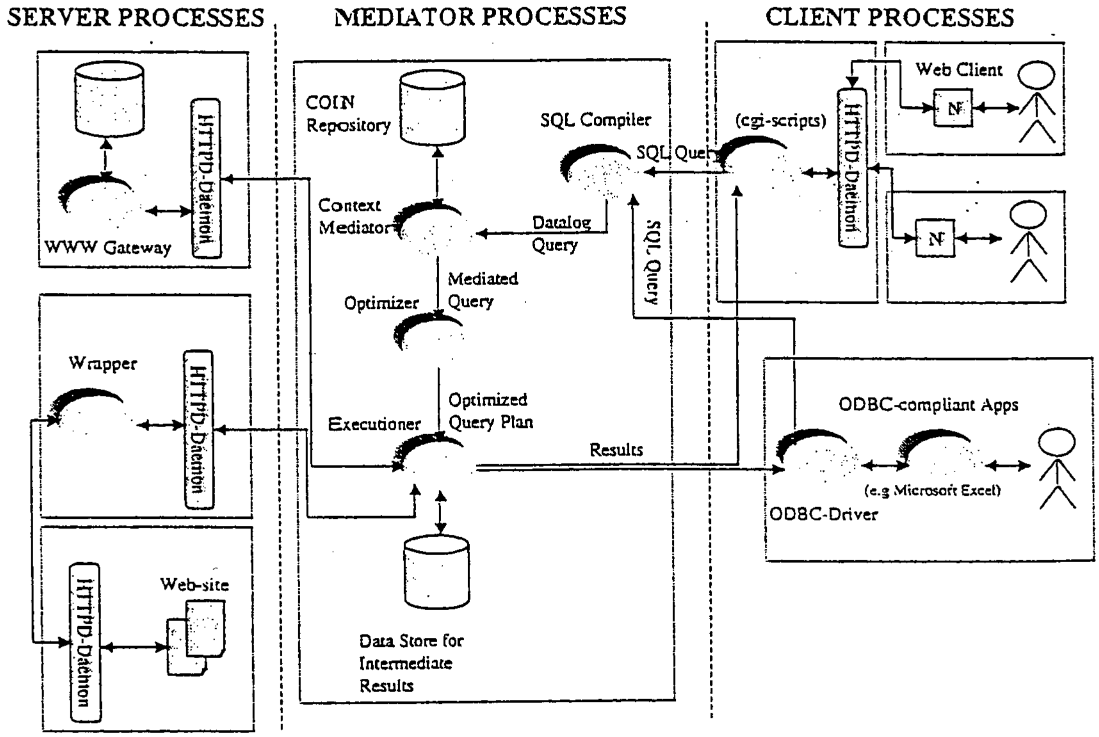


Figure 3: COIN System Overview

4.1 Domain Model and Context definition

The first thing that we need to do is specify the domain model for the domain that we are working in. A *domain model* specifies the semantics of the "types" of information units, which constitutes a common vocabulary used in capturing the semantics of data in disparate sources. In other words it defines the ontology which will be used. The various semantic types, the type hierarchy, and the type signatures (for attributes and modifiers) are all defined in the domain model. Types in the generalized hierarchy are rooted to system types, i.e. types native to the underlying system such as integers, strings, real numbers etc.

Figure 4 depicts part of the domain model that is used in our example. In the domain model described, there are three kinds of relationships expressed.

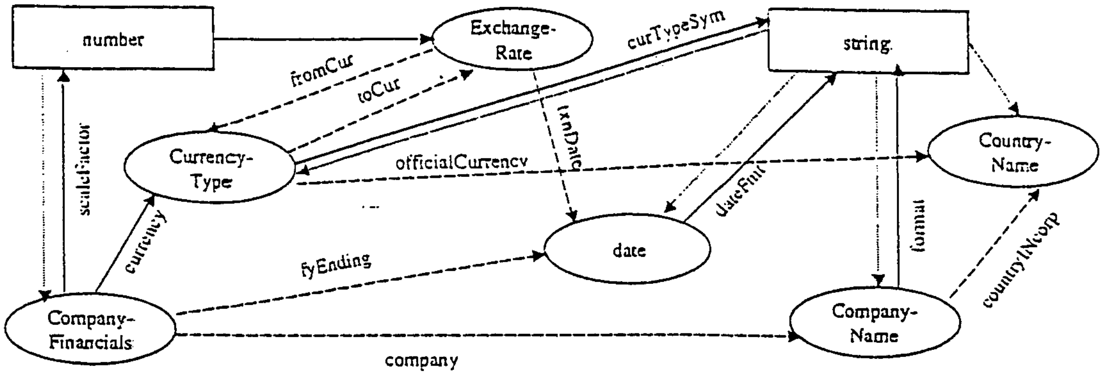


Figure 4:Financial Domain Model

- **Inheritance:** This is the classic type inheritance relationship. All semantic types inherit from basic system types. In the domain model, type `companyFinancials` inherits from basic type string.
- **Attributes:** In COIN [17], objects have two forms of properties, those which are structural properties of the underlying data source and those that encapsulate the underlying assumptions about a particular piece of data. *Attributes* access structural properties of the semantic object in question. For instance, the semantic type `companyfinancials` has two attributes, `company` and `fyEnding`. Intuitively, these attributes define a relationship between objects of the corresponding semantic types. Here, the relationship formed by the `company` attribute states that for any company financial in question, there must be corresponding company to which that company financial belongs. Similarly, the `fyEnding` attribute states that every company financial object has a date when it was recorded.
- **Modifiers:** These define a relationship between semantic objects of the corresponding semantic types. The difference though is that the values of the semantic objects defined by the modifiers have varying interpretations depending on the context. Looking once again at the domain model, the semantic type `companyFinancials` defines two modifiers, `scaleFactor` and `currency`. The value of the object returned by the modifier `scaleFactor` depends on a given context.

Once we have defined the domain model, we need to define the contexts for all the sources. In our case, we have several data sources with the assumptions about their data in figure 5.

A simplified view of what the context might be for the *Worldscope* data source is:

```
modifier(companyFinancials, O, scaleFactor, c_ws, M):-
      cste(basic, M, c_ws, 1000).
modifier(companyFinancials, O, currency, c_ws, M):-
      cste(currencyType, M, c_ws, "USD").
modifier(date, O, dateFmt, c_ws, M):-
      cste(basic, M, c_ws, "American Style /").
```

Datasource	Scale Factor	Currency	Date Format
Worldscope	1000	USD	American "/"
Disclosure	1	Local	American "/"
datastream	1000	Local	European "-"
Olsen	1	Local	European "/"
Quote	1	USD	American "/"

Figure 5: Context Table

Each statement refers to a potential conflict that needs to be resolved by the system. Yet another way to look at it is that each statement corresponds to a modifier relation in the actual domain model. From the domain model shown in Figure 4, we notice that the object *CompanyFinancials* has two modifiers, *scaleFactor* and *currency*. Correspondingly, the first two statements define these two modifiers. Looking at the context table in Figure 5, we notice that the value of the *scaleFactor* in the *Worldscope* context is 1000. The first statement represents that fact. It states that the modifier *scaleFactor* for the object *O* of type *companyFinancials* in the context *c_ws* is the object *M* where (the second line) the object *M* is a constant (*cste*) of type *basic* and has a value of 1000 in the context *c_ws*. In the case of the *Worldscope* data source, all the financial amounts have a scale factor of 1000. That means that in order to get the actual amount of total assets, we will have to multiply the amount returned from the data source by 1000. The next clause determines the *currency* to be in *USD* (i.e., US dollars). The last clause tells the system that the format of the date string in the *Worldscope* is of type American Style with "/" as the delimiting character (mm/dd/yy).

One last thing that needs to be provided as part of a context is the set of conversion functions between different contexts. An example is the conversion between scale factors in different contexts. Following is the conversion routine that is used when scale factors are not equal. The function states that in order to perform conversion of the modifier *scaleFactor* for the object *_O* of semantic type *companyFinancials* in the context *Ctxt* where the modifier value in the source is *Mvs* and the object *_O's* value in the source context is *Vs* and the modifier value in the target context is *Mvt* and the object *_O's* value in the target context is *Vt*, we first find out the *Ratio* between the modifier value in the source context and the modifier value in the target context. We then determine the Object's value in the target context by multiplying its value in the source context with the *Ratio*. *Vt* now contains the appropriately scaled value for the object *_O* in the target context. Note that these conversion rules are defined independent of any specific source or receiver context, the Context Mediator determines if or when such a conversion is needed.

```
cvt(companyFinancials, _O, scaleFactor, Ctxt,
  Mvs, Vs, Mvt, Vt) :-
      Ratio is Mvs / Mvt,
      Vt is Vs * Ratio.
```

4.2 Elevation Axioms

The mapping of data and data-relationships from the sources to the domain model is accomplished via the elevation axioms. There are three distinct operations that define the elevation axioms:

- Define a virtual semantic relation corresponding to each extensional relation.
- Assign to each semantic object defined its value in the context of the source.
- Map the semantic objects in the semantic relation to semantic types defined in the domain model and make explicit any implicit links (attribute initialization) represented by the semantic relation.

We will use the example of the relation Worldscope to show how the relation is elevated. The Worldscope relation is a table in an Oracle database and has the following columns:

Name	Type
COMPANY_NAME	VARCHAR2(80)
LATEST_ANNUAL_FINANCIAL_DATE	VARCHAR2(10)
CURRENT_OUTSTANDING_SHARES	NUMBER
NET_INCOME	NUMBER
SALES	NUMBER
COUNTRY_OF_INCORP	VARCHAR2(40)
TOTAL_ASSETS	NUMBER

And here is what part of the elevated relation looks like:

```
'WorldcAF_p'(
      skolem(companyName, Name, c_ws, 1, 'WorldcAF'( Name, FYEnd, Shares, Income,
Sales, Assets, Incorp)),
      skolem(date, FYEnd, c_ws, 2, 'WorldcAF'( Name, FYEnd, Shares, Income, Sales,
Assets, Incorp)),
        skolem(basic, Shares, c_ws, 3, 'WorldcAF'( Name, FYEnd, Shares, Income, Sales,
Assets, Incorp)),
        skolem(companyFinancials, Income, c_ws, 4, 'WorldcAF'( Name, FYEnd, Shares,
Income, Sales, Assets, Incorp)),
        skolem(companyFinancials, Sales, c_ws, 5, 'WorldcAF'( Name, FYEnd, Shares,
Income, Sales, Assets, Incorp)),
        skolem(companyFinancials, Assets, c_ws, 6, 'WorldcAF'( Name, FYEnd, Shares,
Income, Sales, Assets, Incorp)),
        skolem(countryName, Incorp, c_ws, 7, 'WorldcAF'( Name, FYEnd, Shares, Income,
Sales, Assets, Incorp))
    )   :- 'WorldcAF'(Name, FYEnd, Shares, Income, Sales, Assets, Incorp).
```

We first define a semantic relation for *Worldscope*. A semantic relation is then defined on the semantic objects in the corresponding relation attributes. The data elements derived from the extensional relation are mapped to semantic objects. These semantic objects define a unique object-id for each data element. In the example above each skolem term defines a unique semantic object corresponding to each attribute of the extensional relation. In addition to mapping each physical relation to a corresponding semantic object, we also define and initialize other relations defined in the domain model. The relations that come under this category are attribute and modifiers.

4.3 Mediation System

In the following sections, we will describe each subsystem. We will use the application scenario of the financial analyst trying to gather information about Daimler Benz Corporation. We will use Query1, as presented in Section 2.1, as an example multi-source query. We then describe the application as it is programmed, explaining the domain and how the context information for various sources is specified. Then we will follow the query as it passes through each subsystem.

Query1 is intended to gather financial data for the Daimler Benz Corporation for the year 1993. We get net assets from the Worldscope data source, net sales from the Datastream data source, net income from the Disclosure data source and the latest quotes from Quote data source, which happens to be the CNN web quote server. We will be asking the query in the Worldscope context (i.e., the result of the query will be returned in the Worldscope context.)

4.3.1 SQL to Datalog Query Compiler

The first step is to parse the SQL into its corresponding datalog form and using the elevation axioms it elevates the data sources into its corresponding elevated data objects. The corresponding datalog for the SQL query above is:

```
answer(total_assets, total_sales, net_income, last) :-
        WorldcAF_p(V27, V26, V25, V24, V23, V22, V21),
        DiscAF_p(V20, V19, V18, V17, V16, V15, V14),
        DStreamAF_p(V13, V12, V11, V10, V9, V8),
        quotes_p(V7, q_last),
        Value(V27, c_ws, V5),
        V5 = "DAIMLER-BENZ AG",
        Value(V13, c_ws, V4),
        V4 = "01/05/94",
        Value(V12, c_ws, V3),
        V5 = V3,
        Value(V20, c_ws, V2),
        V5 = V2,
        Value(V7, c_ws, V1),
        V5 = V1,
        Value(V22, c_ws, total_assets),
        Value(V17, c_ws, total_sales),
        Value(V11, c_ws, net_income),
        Value(q_last, c_ws, last).
```

As can be seen, the query now contains elevated data sources along with a set of predicates that map each attribute to its value in the corresponding context. Since the user asked the query in the Worldscope context (denoted by *c_ws*), the last four predicates in the translated query ascertain that the actual values returned as the solution of the query need to be in the Worldscope context. The resulting unmediated datalog query is then fed to the mediation engine.

4.3.2 Mediation Engine

The mediation engine is the part of the system that detects and resolves possible semantic conflicts. In essence, the mediation is a query rewriting process. The actual mechanism of mediation is based on an Abduction Engine [13]. The engine takes a datalog query and a set of domain model axioms and computes a set of abducted queries such that the abducted queries have all the differences resolved. The system does that by incrementally testing for potential semantic conflicts and introducing conversion functions for the resolution of those conflicts. The mediation engine as its output produces a set of queries that take into account all the possible cases given the various conflicts. Using the above example and with the domain model and contexts stated above, we would get the set of abducted queries shown below:

```
answer(V108, V107, V106, V105) :-
        datexform(V104, "European Style -", "01/05/94", "American Style /"),
        Name_map_Dt_Ws(V103, "DAIMLER-BENZ AG"),
        Name_map_Ds_Ws(V102, "DAIMLER-BENZ AG"),
        Ticker_Lookup2("DAIMLER-BENZ AG", V101, V100),
        WorldcAF("DAIMLER-BENZ AG", V99, V98, V97, V96, V108, V95),
        DiscAF(V102, V94, V93, V92, V91, V90, V89),
        V107 is V92 * 0.001,
        Currencytypes(V89, USD),
        DStreamAF(V104, V103, V106, V88, V87, V86),
        Currency_map(USD, V86),
        quotes(V101, V105).

answer(V85, V84, V83, V82) :-
        datexform(V81, "European Style -", "01/05/94", "American Style /"),
        Name_map_Dt_Ws(V80, "DAIMLER-BENZ AG"),
        Name_map_Ds_Ws(V79, "DAIMLER-BENZ AG"),
        Ticker_Lookup2("DAIMLER-BENZ AG", V78, V77),
        WorldcAF("DAIMLER-BENZ AG", V76, V75, V74, V73, V85, V72),
        DiscAF(V79, V71, V70, V69, V68, V67, V66),
        V84 is V69 * 0.001,
        Currencytypes(V66, USD),
        DStreamAF(V81, V80, V65, V64, V63, V62),
        Currency_map(V61, V62),
        <>(V61, USD),
        datexform(V60, "European Style /", "01/05/94", "American Style /"),
        olsen(V61, USD, V59, V60),
        V83 is V65 * V59,
        quotes(V78, V82).

answer(V58, V57, V56, V55) :-
        datexform(V54, "European Style -", "01/05/94", "American Style /"),
        Name_map_Dt_Ws(V53, "DAIMLER-BENZ AG"),
        Name_map_Ds_Ws(V52, "DAIMLER-BENZ AG"),
        Ticker_Lookup2("DAIMLER-BENZ AG", V51, V50),
        WorldcAF("DAIMLER-BENZ AG", V49, V48, V47, V46, V58, V45),
        DiscAF(V52, V44, V43, V42, V41, V40, V39),
        V38 is V42 * 0.001,
        Currencytypes(V39, V37),
        <>(V37, USD),
        datexform(V36, "European Style /", V44, "American Style /"),
        olsen(V37, USD, V35, V36),
        V57 is V38 * V35,
        DStreamAF(V54, V53, V56, V34, V33, V32),
        Currency_map(USD, V32),
        quotes(V51, V55).

answer(V31, V30, V29, V28) :-
        datexform(V27, "European Style -", "01/05/94", "American Style /"),
        Name_map_Dt_Ws(V26, "DAIMLER-BENZ AG"),
        Name_map_Ds_Ws(V25, "DAIMLER-BENZ AG"),
        Ticker_Lookup2("DAIMLER-BENZ AG", V24, V23),
        WorldcAF("DAIMLER-BENZ AG", V22, V21, V20, V19, V31, V18),
        DiscAF(V25, V17, V16, V15, V14, V13, V12),
        V11 is V15 * 0.001,
        Currencytypes(V12, V10),
        <>(V10, USD),
        datexform(V9, "European Style /", V17, "American Style /"),
        olsen(V10, USD, V8, V9),
        V30 is V11 * V8,
DStreamAF(V27, V25, V7, V6, V5, V4),
Currency_map(V3, V4),
<>(V3, USD),
datexform(V2, "European Style /", "01/05/94", "American Style /"),
olsen(V3, USD, V1, V2),
V29 is V7 * V1,
quotes(V24, V28).
```

The mediated query contains four sub-queries. Each of the sub-queries accounts for a potential semantic conflict. For example, the first sub-query deals with the case when there is no currency conversion conflict (i.e., source and receiver use same currency). While the second sub-query takes into account the possibility of currency conversion. Resolving the conflicts may sometime require introducing intermediate data sources. Figure 5 listed some of the context differences in the various data sources that we use for our example. Looking at the table, we observe that one of the possible conflicts is different data sources using different currencies. In order to resolve that difference, the mediation engine has to introduce an intermediary data source. The source used for this purpose is a currency conversion web site (*http://www.oanda.com*) and is referred to as *olsen*. In order to resolve the currency conflict in the second sub-query, the *olsen* source is used to convert the currency to correctly represent data in the currency specified as of the specified date in the specified context. Note that it is the mediator, using the context knowledge, that determines that currency conversion was needed in this case.

4.3.3 Query Planner and optimizer

The query planner module takes the set of datalog queries produced by the mediation engine and produces a query plan. It ensures that an executable plan exists which will produce a result that satisfies the initial query. This is necessitated by the fact that there are some sources that impose restrictions on the type of queries that they can service. In particular, some sources may require that some of the attributes must always be bounded while making queries to those sources. Another limitation sources might have is the kinds of operators that they can handle. One example is that most web sources do not provide an interface that supports all the SQL operators, or they might require that some attributes in queries be always bound. Once the planner ensures than an executable plan exists, it generates a set of constraints on the order in which the different sub-queries can be executed. Under these constraints, the optimizer applies standard optimization heuristics to generate the query execution plan. The query execution plan is essentially an algebraic operator tree in which each operation is represented by a node. There are two types of nodes:

- Access Nodes: Access nodes represent access to remote data sources. Two subtypes of access nodes are:
  - *sfw Nodes:* These nodes represent access to data-sources that do not require input bindings from other sources in the query.
  - *join-sfw Node:* These node have a dependency in that they require input from other data sources in the query. Thus these nodes have to come after the nodes that they depend on while traversing the query plan tree.
- Local Nodes: These nodes represent local operations in local execution engine. There are four subtypes of local nodes.
  - *Join Node:* Joins two trees
  - *Select Node:* This node is used to apply conditions to intermediate results.
  - *CVT Node:* This node is used to apply conversion functions to intermediate query result.
  - *Union Node:* This node represents a union of the results obtained by executing the sub-nodes.

Each node carries additional information about what data-source to access (if it is an access node) and other information that is used by the runtime engine. Some of the information that is carried in each node is a list of attributes in the source and their relative position, list of condition operations and any literals and other information like the conversion formula in the case of a conversion node. The query plan for the first sub-query of the mediated query is shown in the Appendix. The query plan that is produced by the planner is then forwarded to the runtime engine.

4.3.4 Runtime engine

The runtime execution engine executes the query plan. Given a query plan, the execution engine traverses the query plan tree in a depth-first manner starting from the root node. At each node, it computes the sub-trees for that node and then applies the operation specified for that node. For each sub-tree the engine recursively descends down the tree until it encounters an access node. For that access node, it composes a SQL query and sends it off to the remote source. The results of that query are then stored in the local store. Once all the sub-trees have been executed and all the results are in the local store, the operation associated with that node is executed and the results collected. This operation continues until the root of the query is reach. At this point the execution engine has the required set of results corresponding to the original query. These results are then sent back to the user and the process is completed.

4.4 Web Wrapper

The original query used in our example, contained access to a quote server to get the most recent quotes for the company in question, i.e. Daimler-Benz. As opposed to the rest of the sources, the quote server that we used is a web quote server. In order to access the web sources such as this one, we have developed a technology that lets users treat web sites as relational data sources. Users can then issue SQL queries just as they would to any relation in the relational domain, thus combining multiple sources and creating queries as the one above. This technology is called *web-wrapping* and we have an implementation for this technology which is called *web wrapping engine* [18]. Using the web wrapper engine (web wrapper for short) the application developers can very rapidly *wrap* a structured or semi-structured web site and export the schema for the users to query against. Once the source has been wrapped, it can be used as a relational source in any query.

4.4.1 Web wrapper architecture

Figure 6 shows the architecture of the web wrapper. The system takes the SQL query as input. It parses the query along with the specifications for the given web site. A query plan is then constituted. The plan constitutes of what web sites to send http requests and what documents on those web sites. The executioner then executes the plan. Once the pages are fetched, the executioner then extracts the required information from the pages and presents that to the user.

4.4.2 Wrapping a web site

For our query, the relation *quote* is actually a web quote server that we access using our web wrapper. In order to *wrap* a site, you need to create a specification file. For each Web page or set of Web pages the generic Web Wrapper engine utilizes a specification file to guide it through the data extractions process. The specification file contains information about the locations on the web for both input and output data. The Web Wrapper Engine utilizes this information during query execution to get information back from the set of Web pages as if it were a relational database. This file is plain text file and contains information such as the exported schema, the URL of the web site to access, and a regular expression that will be used to extract the actual information from the web page. In our example we use the *CNN* quote server to get quotes.

A simplified specification file is included below:

```
#HEADER
#RELATION=quotes
#HREF=GET http://qs.cnnfn.com
#EXPORT= quotes.Cname quotes.Last
#ENDHEADER

#BODY
#PAGE
#HREF = POST http://qs.cnnfn.com/cgibin/stockquote?
  symbols=##quotes.Cname##
#CONTENT=last:&nbsp </font><FONT
  SIZE=+1><b>##quotes.Last##</FONT></TD>
#ENDPAGE
#ENDBODY
```

The specification has two parts, *Header* and *Body*. The *Header* part specifies information about the name of the relation and the exported schema. In the above case, the schema that we decided to export has two attributes, *Cname*, the ticket of the company and *Last* the latest quote. The *Body* Portion of the file specifies how to actually access the page (as defined in the *HREF* field) and what regular expression to use (as defined in the *CONTENT* field). Once the specification file is written and placed where the web wrapper can read it, we are ready to use the system. We can start making queries against the new relation that we just created.

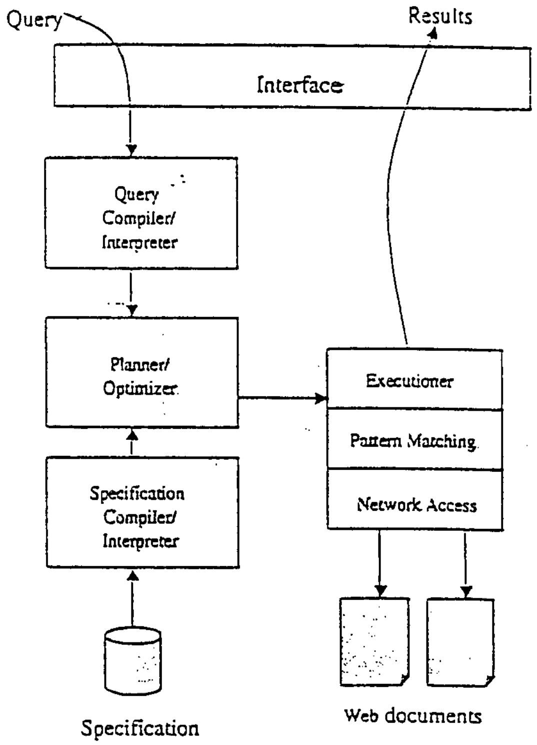


Figure 6: Web Wrapper Architecture

4.4.3 Web Wrapping and XML

The eXtensible Markup Language (XML) for Web pages is becoming increasingly accepted. This provides opportunities for the Web Wrapping Engine, in particular, and the Context Interchange System, in general. First, the XML tags provide a much easier and explicit demarcation of the location of fields within a web page. That makes the extraction of data much simpler for the Web Wrapper. On the other hand, XML is primarily a syntactic facility. You may have a tag <PRICE>, but XML does not provide the semantics, such as what is the currency of the price, does it include tax, does it include shipping costs, etc. The Context Interchange approach is the next step in the evolution of XML and the Web to provide the semantics that is so critical to the effective exchange of information.

5. Conclusions

In this paper, we have described a novel approach to the problem of resolving semantic differences between disparate information sources by automatically detecting and resolving semantic conflicts between those sources based on the knowledge of the contexts of those data sources in a particular domain. We have also described and explained the architecture and implementation of the prototype, and discussed the prototype at work by using an example scenario. More details pertaining to this scenario and a demonstration of its operation can be found at http://context.mit.edu/~coin/demos/tasc/saved/queries/q11.html.

Acknowledgements

Work reported herein has been supported, in part, by the Advanced Research Projects Agency (ARPA) and the USAF/Rome Laboratory under contract F30602-93-C-0160, TASC, Inc. (and its parent, PRIMARK Corporation), the MIT PROductivity From Information Technology (PROFIT) Program, and the MIT Total Data

Quality Management (TDQM) Program. Information about the Context Interchange project can be obtained at `http://context.mit.edu/~coin`.

Bibliography

[1] Adil Daruwala, Cheng Goh, Scot Hofmeister, Karim Husein, Stuart Madnick, Michael Siegel. "The Context Interchange Network Prototype", Center For Information System Laboratory (CISL), working paper, #95-01, Sloan School of Management, Massachusetts Institute of Technology, February 1995

[2] Goh, C., Madnick, S., Siegel, M. "Context Interchange: Overcoming the challenges of Large-scale interoperable database systems in a dynamic environment". Proc. of Intl. Conf. On Information and Knowledge Management. 1994.

[3] Arens, Y. and Knobloch, C. "Planning and reformulating queries for semantically-modeled multidatabase". Proc. of the Intl. Conf. on Information and Knowledge Management. 1992

[4] Garcia-Molina, H. "The TSIMMIS Approach to Mediation: Data Models and Languages". Proc. of the Conf. on Next Generation Information Technologies and Systems. 1995.

[5] Tomasic, A. Rashid, L., and Valduriez, P. "Scaling Heterogeneous databases and the design of DISCO". Proc. of the Intl. Conf. on Distributed Computing Systems. 1995.

[6] Levy, A., Srivastava, D. and Krik, T. "Data Model and Query Evaluation in Global Information Systems". Journal of Intelligent Information Systems. 1995.

[7] Papakonstantinou, Y., Gupta, a., and Haas, L. "Capabilities-Based Query Rewriting in Mediator Systems". Proc. of the 4th Intl. Conf. on Parallel and Distributed Information Systems. 1996.

[8] Duschka, O., and Genesereth, M. "Query Planning in Infomaster". http://infomaster.standord.edu. 1997.

[9] Wiederhold, G. "Mediation in the Architecture of Future Information Systems". Computer, 23(3), 1992.

[10] Kifer, M., Lausen, G., and Wu, J. Logical Foundations of Object-oriented and Frame-based Languages. JACM 5 (1995), pp. 741-843.

[11] Pena, F. PENNY: A Programming Language and Compiler for the Context Interchange Project. CISL Working Paper #97-06, 1997

[12] McCarthy, J. Generality in Artificial Intelligence. Communications of the ACM 30, 12(1987), pp. 1030-1035.

[13] KaKas, A. C., Kowalski, R. A. and Toni, F. Abductive Logic Programming. Journal of Logic and Computation 2, 6 (1993), pp. 719-770.

[14] Sheth, A. P., and Larson, J. A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 22, 3 (1990), pp. 183-236

[15] Jakobiasik, M. Programming the web-design and implementation of a multidatabase browser. Technical Report, CISL, WP #96-04, 1996

[16] Qu, J. F. Data wrapping on the world wide web. Technical Report, CISL WP #96-05, 1996

[17] Goh, C. H. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology, 1996.

[18] Bressan, S., Bonnet, P. Extraction and Integration of Data from Semi-structured Documents into Business Applications. To be published.

[19] Madnick, S. Metadata Jones and the Tower of Babel: The Challenge of Large-Scale Heterogeneity. Proceedings of the IEEE Meta-data Conference, April 1999.

APPENDIX: Part of the Query Execution Plan

(the entire plan can be found at http://context.mit.edu/~coin/demos/tasc/saved/saved-results/q11_t3.html)

```
SELECT
 ProjL: [att(1, 5), att(1, 4), att(1, 3), att(1, 2)]
 CondL: [att(1, 1) = "01/05/94"]
  JOIN-SFW-NODE DateXform
  ProjL: [att(1, 3), att(2, 4), att(2, 3), att(2, 2), att(2,1)]
  CondL: [att(1, 1) = att(2, 5), att(1, 2) = "European Style
  -", att(1, 4) = "American Style /"]
   CVT-NODE 'V18' is 'V17' * 0.001
   Projl: [att(2, 1), att(1, 1), att(2, 2), att(2, 3), att(2, 4)]
   Condl: ['V17' = att(2, 5)]
    JOIN-SFW-NODE quotes
    ProjL: [att(2, 1), att(2, 2), att(1, 2), att(2, 3), att(2, 4)]
    CondL: [att(1, 1) = att(2, 5)]

     JOIN-NODE
```

```
ProjL: [att(2, 1), att(2, 2), att(2, 3), att(2, 4), att(2, 5)]
CondL: [att(1, 1) = att(2, 6)]
 SELECT
 ProjL: [att(1, 2)]
 CondL: [att(1, 1) = 'USD']
  SFW-NODE Currency_map
  ProjL: [att(1, 1), att(1, 2)]
  CondL: []

 JOIN-NODE
 ProjL: [att(2, 1), att(1, 2), att(1, 3),att(2, 2),att(2, 3),att(1, 4)]
 CondL: [att(1, 1) = att(2, 4)]
  SFW-NODE Datastream
  ProjL: [att(1, 2), att(1, 3), att(1, 1), att(1, 6)]
  CondL: []

  JOIN-NODE
  ProjL: [att(2, 1), att(2, 2), att(2, 3), att(2, 4)]
  CondL: [att(1, 1) = att(2, 5)]
   SELECT
   ProjL: [att(1, 2)]
   CondL: [att(1, 1) = 'USD']
    SFW-NODE Currencytypes
    ProjL: [att(1, 2), att(1, 1)]
    CondL: []

   JOIN-NODE
   ProjL: [att(2, 1), att(1, 2), att(2, 2), att(2, 3), att(1, 3)]
   CondL: [att(1, 1) = att(2, 4)]
    SFW-NODE Disclosure
    ProjL: [att(1, 1), att(1, 4), att(1, 7)]
    CondL: []

    JOIN-NODE
    ProjL: [att(1, 1), att(2, 1), att(2, 2), att(2, 3)]
    CondL: []
     SELECT
     ProjL: [att(1, 2)]
     CondL: [att(1, 1) = "DAIMLER-BENZ AG"]
      SFW-NODE Worldscope
      ProjL: [att(1, 1), att(1, 6)]
      CondL: []

     JOIN-NODE
     ProjL: [att(1, 1), att(2, 1), att(2, 2)]
     CondL: []
      SELECT
      ProjL: [att(1, 2)]
      CondL: [att(1, 1) = "DAIMLER-BENZ AG"]
       SFW-NODE Ticker_Lookup2
       ProjL: [att(1, 1), att(1, 2)]
       CondL: []

      JOIN-NODE
      ProjL: [att(2, 1), att(1, 1)]
      CondL: []
       SELECT
       ProjL: [att(1, 2)]
       CondL: [att(1, 1) = "DAIMLER-BENZ AG"]
        SFW-NODE Name_map_Ds_Ws
        ProjL: [att(1, 2), att(1, 1)]
        CondL: []

       SELECT
       ProjL: [att(1, 2)]
       CondL: [att(1, 1) = "DAIMLER-BENZ AG"]
        SFW-NODE Name_map_Dt_Ws
        ProjL: [att(1, 2), att(1, 1)]
        CondL: []
```

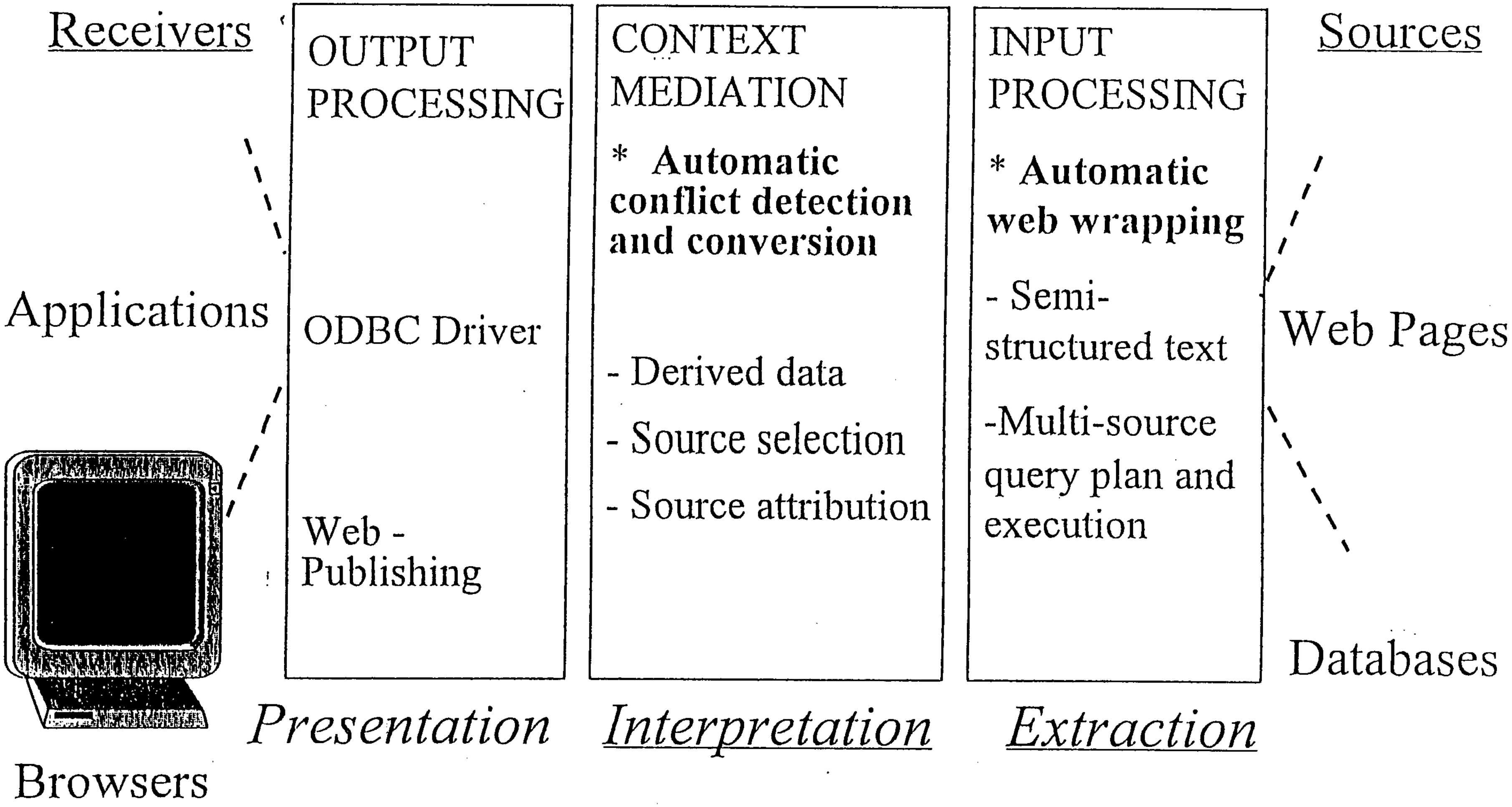
4. DEMONSTRATIONS

Context Interchange (COIN) Project

June 1, 1999

Stuart Madnick
Michael Siegel

Research Overview



Receivers

Applications

Browsers

OUTPUT PROCESSING

ODBC Driver

Web - Publishing

CONTEXT MEDIATION

* **Automatic conflict detection and conversion**

- Derived data
- Source selection
- Source attribution

INPUT PROCESSING

* **Automatic web wrapping**

- Semi-structured text

-Multi-source query plan and execution

Sources

Web Pages

Databases

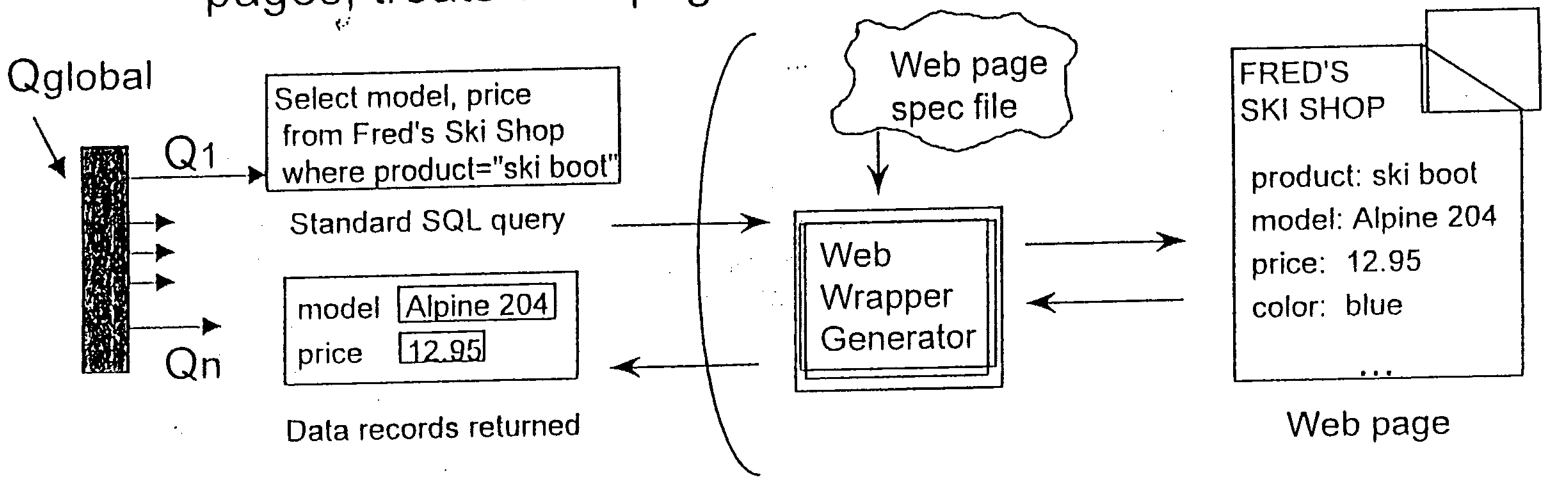
Presentation

Interpretation

Extraction

TECHNOLOGIES

1. Web Wrapper Generator (Extracts data from Web pages, treats Web pages as a relational database)



Qglobal

Q1

Qn

Select model, price
from Fred's Ski Shop
where product="ski boot"

Standard SQL query

model	Alpine 204
price	12.95

Data records returned

Web page spec file

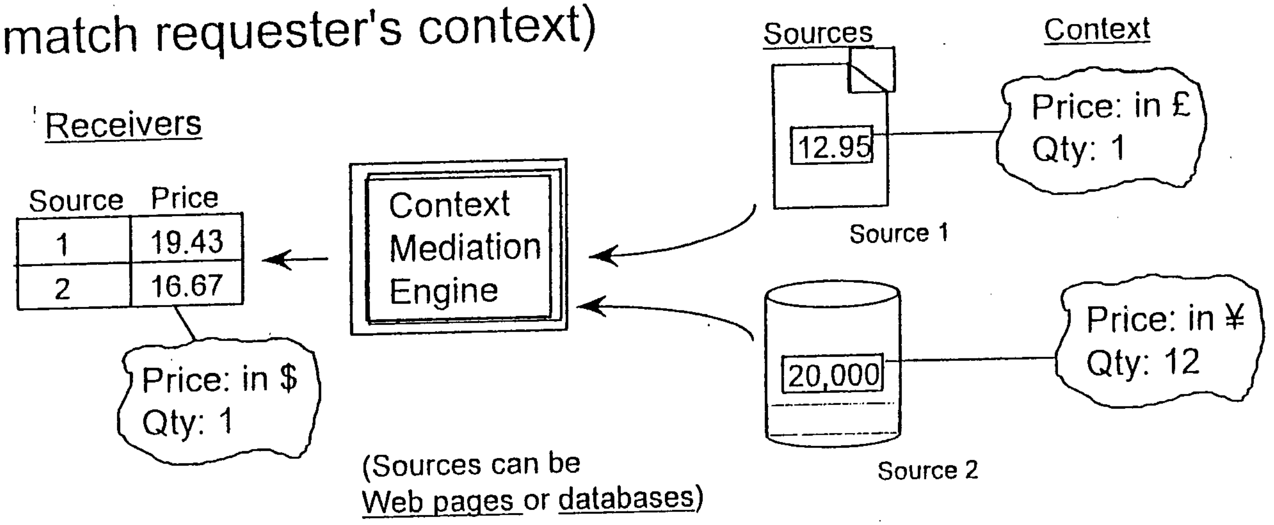
Web Wrapper Generator

FRED'S
SKI SHOP

product: ski boot
model: Alpine 204
price: 12.95
color: blue
...

Web page

2. Context Mediation Engine (Converts context of data to match requester's context)



Receivers

Source	Price
1	19.43
2	16.67

Price: in $
Qty: 1

Context Mediation Engine

Sources

12.95

Source 1

20,000

Source 2

Context

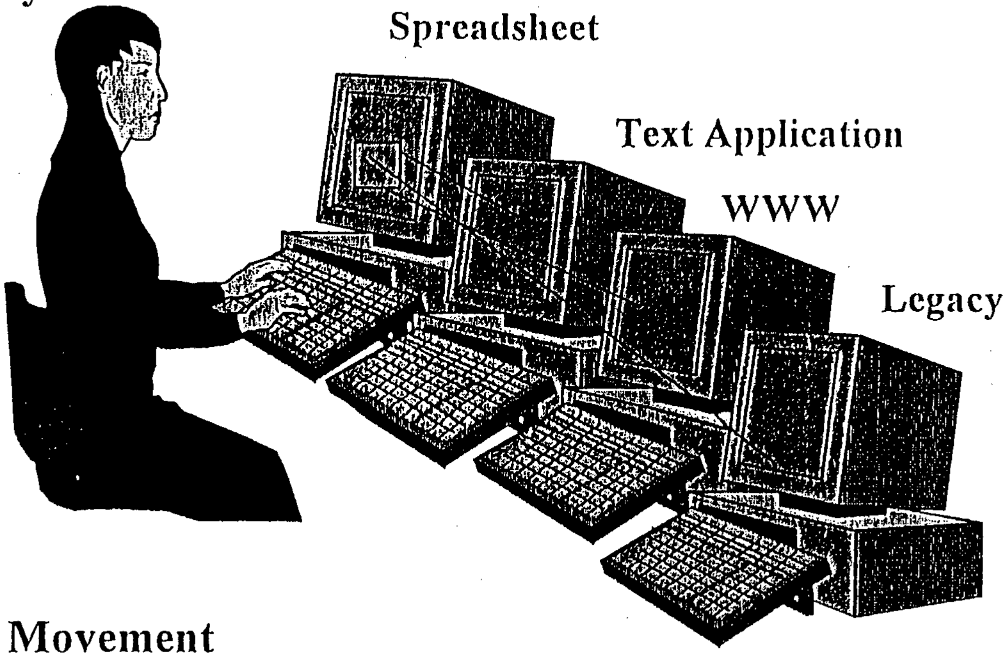
Price: in £
Qty: 1

Price: in ¥
Qty: 12

(Sources can be
Web pages or databases)

Querying Many Sites

Example Project 1 : HTML



Research Analyst
or
Trader

Spreadsheet

Text Application

WWW

Legacy Application

Manual Data Movement

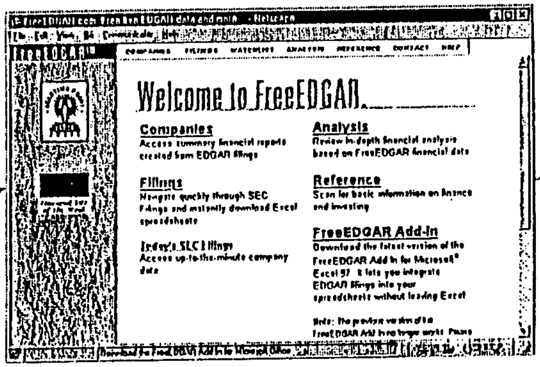
Web-Wrapper Technology

SQL Query

Select Edgar.Net_income
From Edgar
Where Edgar.Ticker=intc
and Edgar.Form=10-Q

Web page spec file

Web Wrapper Generator


FreeEDGAR.com Free EDGAR data and more - Netscape

File Edit View Go Communicator Help
COMPANIES FILINGS WATCHLIST ANALYSIS REFERENCE CONTACT HELP
FreeEDGAR

Welcome to FreeEDGAR

Companies
Access summary financial reports created from EDGAR filings

Filings
Navigate quickly through SEC filings and instantly download Excel spreadsheets

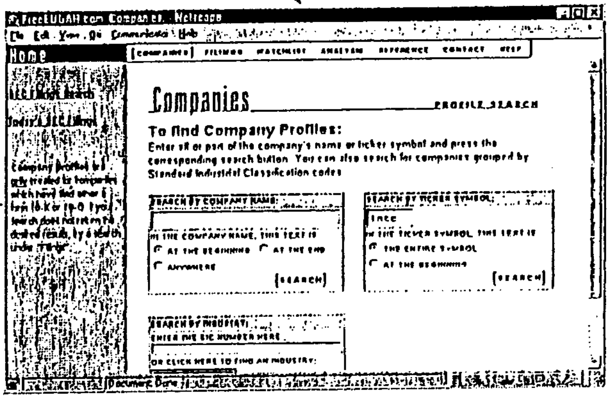
Today's SEC Filings
Access up-to-the-minute company data

Analysis
Review in-depth financial analysis based on FreeEDGAR financial data

Reference
Scan for basic information on finance and investing

FreeEDGAR Add-In
Download the latest version of the FreeEDGAR Add-In for Microsoft® Excel 97. It lets you integrate EDGAR filings into your spreadsheets without leaving Excel

Note: The previous version of the FreeEDGAR Add-In no longer works. Please


FreeEDGAR.com Companies - Netscape

File Edit View Go Communicator Help
Home
[COMPANIES] FILINGS WATCHLIST ANALYSIS REFERENCE CONTACT HELP
PROFILE SEARCH
[SEARCH]
[SEARCH]

Companies

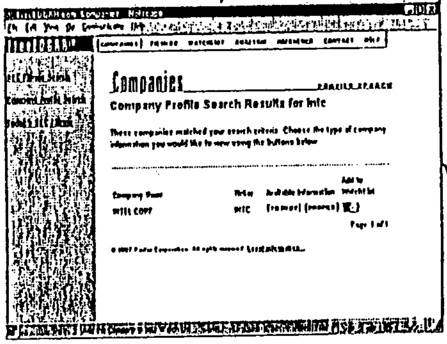
To find Company Profiles:
Enter all or part of the company's name or ticker symbol and press the corresponding search button. You can also search for companies grouped by Standard Industrial Classification codes

SEARCH BY COMPANY NAME:
IN THE COMPANY NAME, THIS TEXT IS
● AT THE BEGINNING ○ AT THE END
○ ANYWHERE

SEARCH BY TICKER SYMBOL:
intc
IN THE TICKER SYMBOL, THIS TEXT IS
● THE ENTIRE SYMBOL
○ AT THE BEGINNING

SEARCH BY INDUSTRY:
ENTER THE SIC NUMBER HERE
OR CLICK HERE TO FIND AN INDUSTRY:

Company profiles are only created for companies which have filed either a form 10-K or 10-Q. If your search does not return the desired results, try a search under Filings.


FreeEDGAR.com Companies - Netscape

File Edit View Go Communicator Help
FreeEDGAR
[COMPANIES] FILINGS WATCHLIST ANALYSIS REFERENCE CONTACT HELP
PROFILE SEARCH

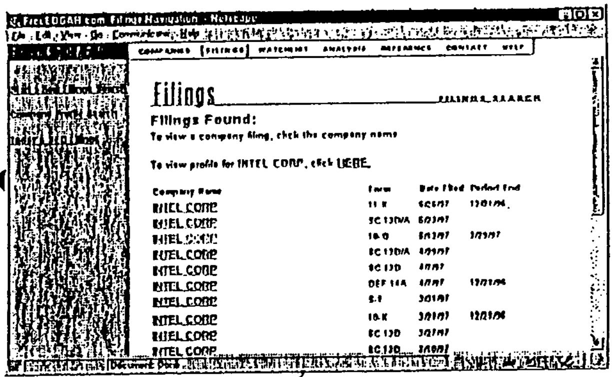
Companies

Company Profile Search Results for intc

These companies matched your search criteria. Choose the type of company information you would like to view using the buttons below

Company Name	Ticker	Available Information	Add to Watchlist
INTEL CORP	INTC	[financials] [filings]	[+]

Page 1 of 1


FreeEDGAR.com Filings Navigation - Netscape

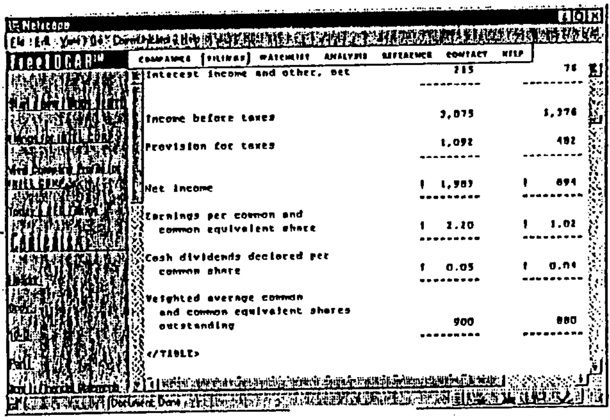
File Edit View Go Communicator Help
COMPANIES [FILINGS] WATCHLIST ANALYSIS REFERENCE CONTACT HELP
FILINGS SEARCH

Filings

Filings Found:
To view a company filing, click the company name

To view profile for INTEL CORP, click HERE.

Company Name	Form	Date Filed	Period End
INTEL CORP	11-K	6/26/97	12/31/96
INTEL CORP	SC 13D/A	6/23/97	
INTEL CORP	10-Q	5/13/97	3/29/97
INTEL CORP	SC 13D/A	4/29/97	
INTEL CORP	SC 13D	4/7/97	
INTEL CORP	DEF 14A	4/7/97	12/28/96
INTEL CORP	S-8	3/31/97	
INTEL CORP	10-K	3/21/97	12/28/96
INTEL CORP	SC 13D	3/27/97	
INTEL CORP	SC 13D	3/10/97	

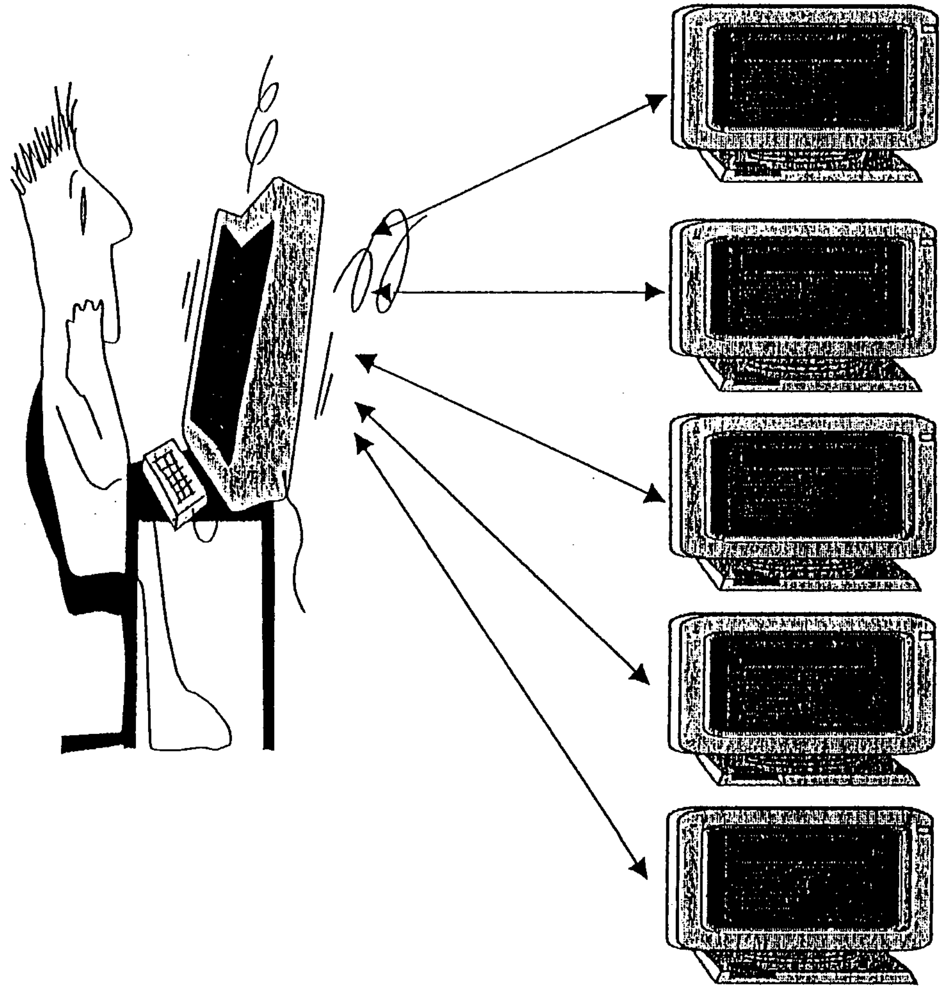

Netscape

File Edit View Go Communicator Help
FreeEDGAR
COMPANIES [FILINGS] WATCHLIST ANALYSIS REFERENCE CONTACT HELP

```
Interest income and other, net                     215         76
                                            ---------  ---------


Income before taxes                              3,075      1,376

Provision for taxes                              1,092        482
                                            ---------  ---------

Net income                                   $   1,983  $     894
                                            =========  =========
Earnings per common and
  common equivalent share                    $    2.10  $    1.02
                                            =========  =========
Cash dividends declared per
  common share                               $    0.05  $    0.04
                                            =========  =========
Weighted average common
  and common equivalent shares
  outstanding                                      900        880
                                            =========  =========
</TABLE>
```

Ticker	Net Income
INTC	1,983

Problem: Providing Integrated Data and Analysis



Equity prices - TIBCO
Real-time feed

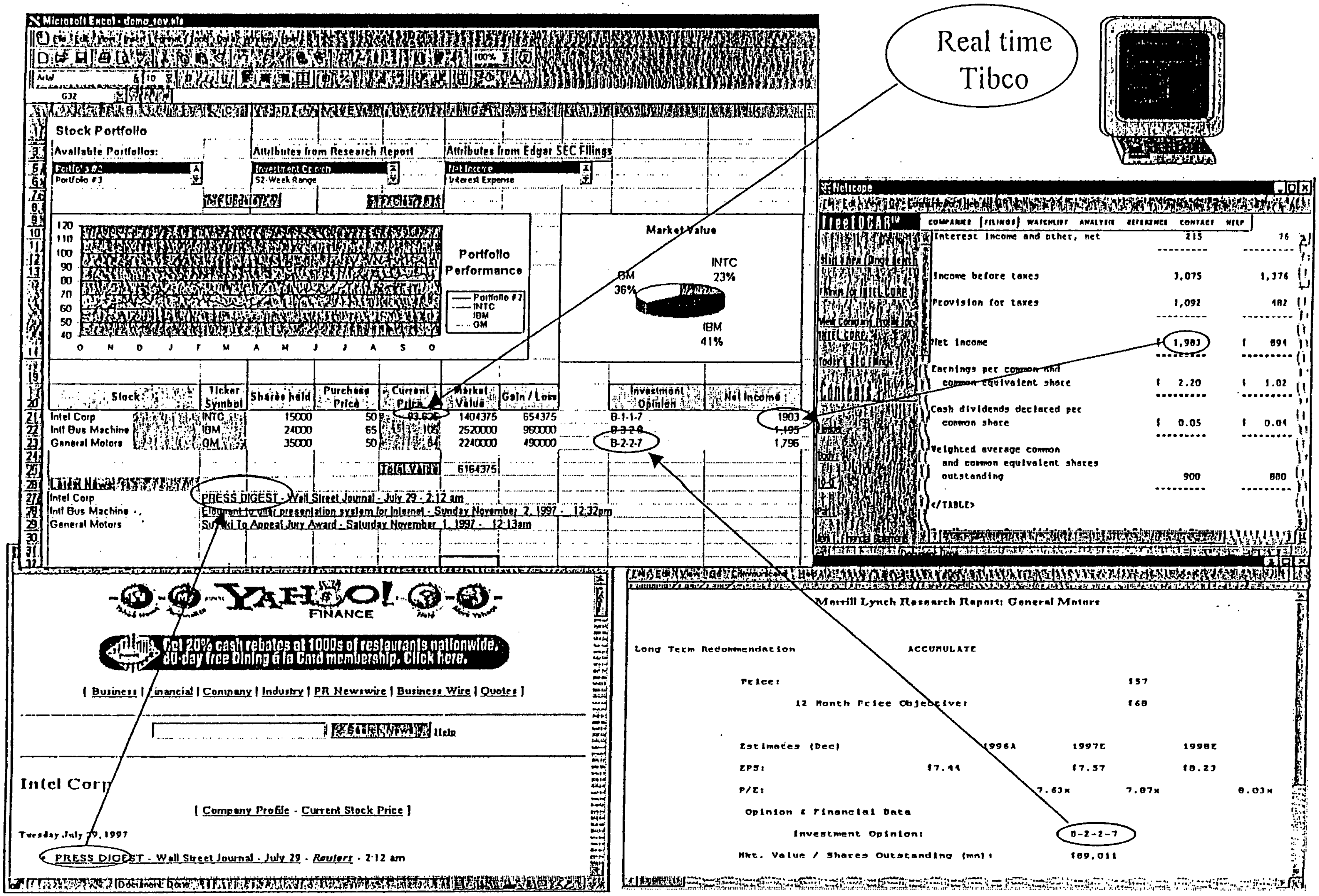
SEC Filings - EDGAR
Web based

News - Reuters, Newswire and Businesswire
Web based

Research Reports
Text based - Intranet

Company Home Page
Market Updates - *Web based*

Spreadsheet Interface



Stock Portfolio

Available Portfolios: Portfolio #2, Portfolio #3

Attributes from Research Report: Investment Opinion, 52-Week Range

Attributes from Edgar SEC Filings: Net Income, Interest Expense

Stock	Ticker Symbol	Shares held	Purchase Price	Current Price	Market Value	Gain / Loss	Investment Opinion	Net Income
Intel Corp	INTC	15000	50	93.625	1404375	654375	B-1-1-7	1903
Intl Bus Machine	IBM	24000	65	105	2520000	960000	~~B-3-2-8~~	~~1,195~~
General Motors	GM	35000	50	64	2240000	490000	B-2-2-7	1,796
				Total Value	6164375			

Latest News:

Intel Corp — PRESS DIGEST - Wall Street Journal - July 29 - 2:12 am

Intl Bus Machine — Element to offer presentation system for Internet - Sunday November 2, 1997 - 12:32pm

General Motors — Suzuki To Appeal Jury Award - Saturday November 1, 1997 - 12:13am

Interest income and other, net	215	76
Income before taxes	3,075	1,376
Provision for taxes	1,092	482
Net income	$ 1,983	$ 894
Earnings per common and common equivalent share	$ 2.20	$ 1.02
Cash dividends declared per common share	$ 0.05	$ 0.04
Weighted average common and common equivalent shares outstanding	900	880

</TABLE>

YAHOO! FINANCE

Get 20% cash rebates at 1000s of restaurants nationwide. 30-day free Dining à la Card membership. Click here.

[Business | Financial | Company | Industry | PR Newswire | Business Wire | Quotes]

Help

Intel Corp

[Company Profile - Current Stock Price]

Tuesday, July 29, 1997

- PRESS DIGEST - Wall Street Journal - July 29 - Reuters - 2:12 am

Merrill Lynch Research Report: General Motors

Long Term Recommendation ACCUMULATE

Price: $57

12 Month Price Objective: $68

Estimates (Dec)	1996A	1997E	1998E
EPS:	$7.44	$7.57	$8.23
P/E:	7.63x	7.87x	8.03x

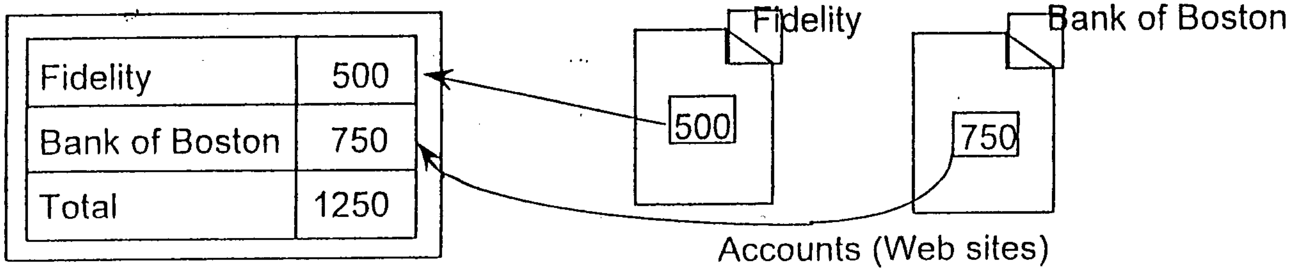
Opinion & Financial Data

Investment Opinion: B-2-2-7

Mkt. Value / Shares Outstanding (mn): $89,011

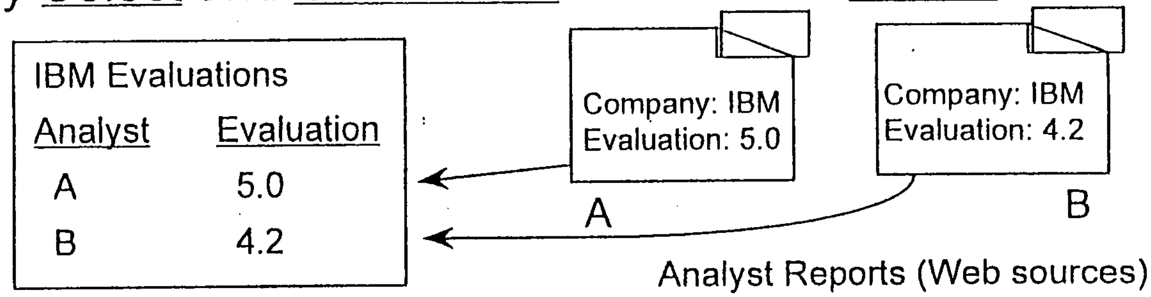
SAMPLE APPLICATIONS

- Automate Extraction of data from specific Web sites into user tool, like Excel, or own Web browser / consolidator



Fidelity	500
Bank of Boston	750
Total	1250

- Automatically Select and Consolidate information across Web sites



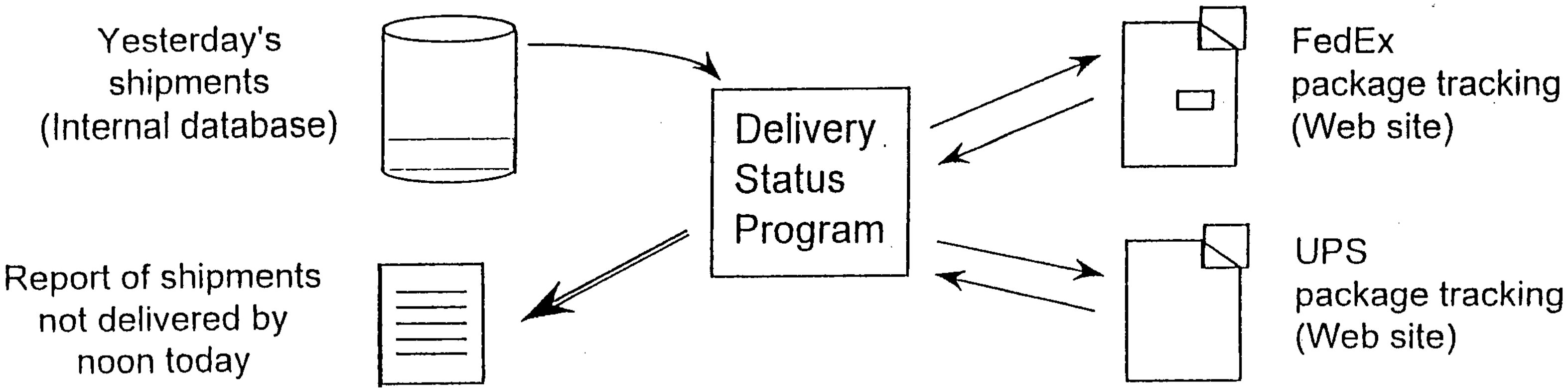
IBM Evaluations

Analyst	Evaluation
A	5.0
B	4.2

Company: IBM
Evaluation: 5.0

Company: IBM
Evaluation: 4.2

- Integrate Internet / Intranet / Client Server networks for internal operations



Yesterday's shipments (Internal database)
Delivery Status Program
FedEx package tracking (Web site)
UPS package tracking (Web site)
Report of shipments not delivered by noon today

Data Interpretation:
The Importance of Context

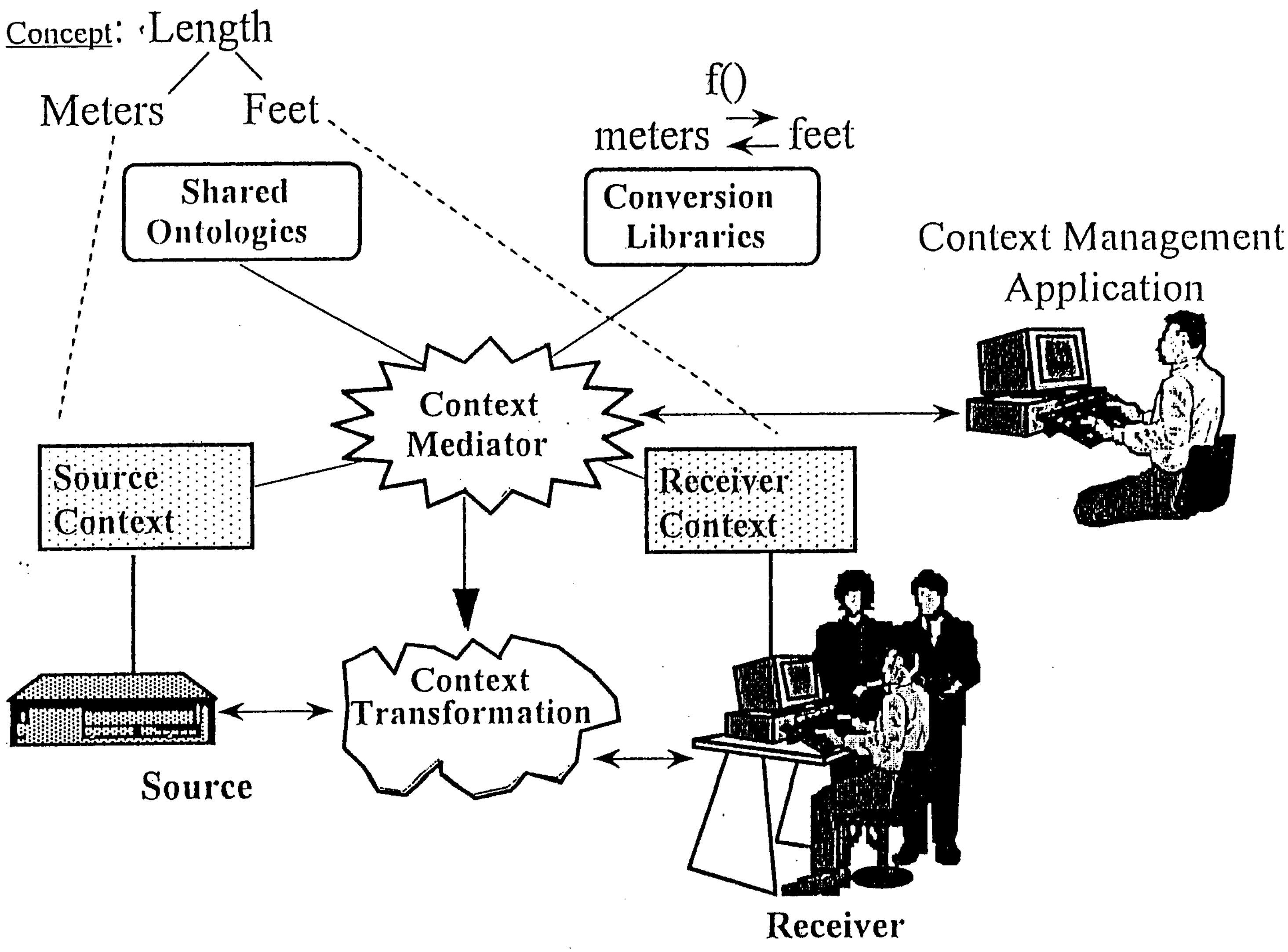
(Information on HONDA)

DISCLOSURE		DATALINE		
ATTRIBUTES	*VALUES*	*VALUES*	*ATTRIBUTES*	
COMPNO	3842	HOND	CODE	*(1), (2)*
CF	19,860,228	28-02-86	PERIOD ENDING	*(1), (3)*
NI	146,502	146,502	EARNED FOR ORDINARY	*(1)*
NRCEX *(ROE)*	0.11	19.57	RETURN ON EQUITY	*(1), (4), (5)*

CONTEXT DIFFERENCES ILLUSTRATED:

(1) Attribute naming
(2) Codes used
(3) Conventions/Format
(4) Scale
(5) Calculation

The Context Interchange Approach



Concept: ·Length

Meters Feet

f()

meters ⇄ feet

Shared Ontologies

Conversion Libraries

Context Management Application

Context Mediator

Source Context

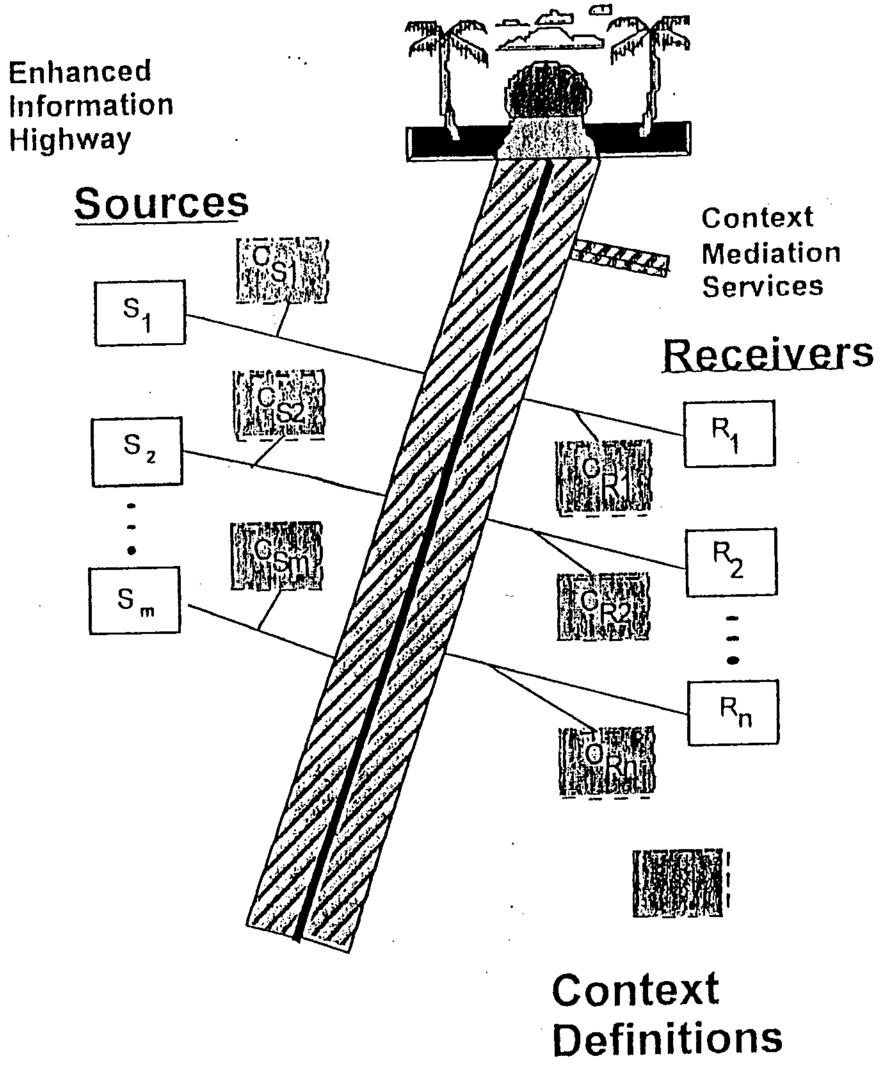
Receiver Context

Context Transformation

Source

Receiver

Context Mediation Services



Technology Transfer

- Global Infotek
- PriceWaterhouseCoopers
- Merrill Lynch

Demonstrations

- http://context.mit.edu/~coin

Future Work

- Practical Context - Context-XML
- Web Wrapper Specification Wizard
- Improve Multi-Datasource Query Planner and Execution Engine
- Develop Practical Applications on the Web

☆U.S. GOVERNMENT PRINTING OFFICE: 1999-510-079-81218

DISTRIBUTION LIST

addresses	number of copies
DR. RAYMOND A. LIUZZI AFRL/IFTD 525 BROOKS ROAD ROME NY 13441-4505	10
MASSACHUSETTS INST. OF TECHNOLOGY SLOAN SCHOOL OF MANAGEMENT 30 WADSWORTH ST. CAMBRIDGE MA 02142	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P.STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

AFRL/HESC-TDC 1
2698 G STREET, BLDG 190
WRIGHT-PATTERSON AFB OH 45433-7604

ATTN: SMDC IM PL 1
US ARMY SPACE & MISSILE DEF CMD
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

TECHNICAL LIBRARY D0274(PL-TS) 1
SPAWARSYSCEN
53560 HULL ST.
SAN DIEGO CA 92152-5001

COMMANDER, CODE 4TL000D 1
TECHNICAL LIBRARY, NAWC-WD
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6100

COR, US ARMY AVIATION & MISSILE CMD 2
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RD-OB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000

REPORT LIBRARY 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

ATTN: D'BORAH HART 1
AVIATION BRANCH SVC 122.10
FOB10A, RM 931
800 INDEPENDENCE AVE, SW
WASHINGTON DC 20591

AFIWC/MSY 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

ATTN: KAROLA M. YOURISON 1
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE
PITTSBURGH PA 15213

USAF/AIR FORCE RESEARCH LABORATORY 1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB MA 01731-3004

ATTN: EILEEN LADUKE/D460 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

OUSD(P)/DTSA/DUTD 1
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

SOFTWARE ENGR'G INST TECH LIBRARY 1
ATTN: MR DENNIS SMITH
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PA 15213-3890

USC-ISI 1
ATTN: DR ROBERT M. BALZER
4676 ADMIRALTY WAY
MARINA DEL REY CA 90292-6695

KESTREL INSTITUTE 1
ATTN: DR CORDELL GREEN
1801 PAGE MILL ROAD
PALO ALTO CA 94304

ROCHESTER INSTITUTE OF TECHNOLOGY 1
ATTN: PROF J. A. LASKY
1 LOMB MEMORIAL DRIVE
P.O. BOX 9887
ROCHESTER NY 14613-5700

AFIT/ENG 1
ATTN:TOM HARTRUM
WPAFB OH 45433-6583

THE MITRE CORPORATION 1
ATTN: MR EDWARD H. BENSLEY
BURLINGTON RD/MAIL STOP A350
BEDFORD MA 01730

ANDREW A. CHIEN 1
SAIC CHAIR PROF (SCI APL INT CORP)
USCD/CSE-AP&M 4808
9500 GILMAN DRIVE, DEPT. 0114
LAJOLLA CA 92093-0114

HONEYWELL, INC. 1
ATTN: MR BERT HARRIS
FEDERAL SYSTEMS
7900 WESTPARK DRIVE
MCLEAN VA 22102

SOFTWARE ENGINEERING INSTITUTE 1
ATTN: MR WILLIAM E. HEFLEY
CARNEGIE-MELLON UNIVERSITY
SEI 2218
PITTSBURGH PA 15213-38990

UNIVERSITY OF SOUTHERN CALIFORNIA 1
ATTN: DR. YIGAL ARENS
INFORMATION SCIENCES INSTITUTE
4676 ADMIRALTY WAY/SUITE 1001
MARINA DEL REY CA 90292-6695

COLUMBIA UNIV/DEPT COMPUTER SCIENCE 1
ATTN: DR GAIL E. KAISER
450 COMPUTER SCIENCE BLDG
500 WEST 120TH STREET
NEW YORK NY 10027

AFIT/ENG 1
ATTN: DR GARY B. LAMONT
SCHOOL OF ENGINEERING
DEPT ELECTRICAL & COMPUTER ENGRG
WPAFB OH 45433-6583

NSA/OFC OF RESEARCH 1
ATTN: MS MARY ANNE OVERMAN
9800 SAVAGE ROAD
FT GEORGE G. MEADE MD 20755-6000

AT&T BELL LABORATORIES 1
ATTN: MR PETER G. SELFRIDGE
ROOM 3C-441
600 MOUNTAIN AVE
MURRAY HILL NJ 07974

ODYSSEY RESEARCH ASSOCIATES, INC. 1
ATTN: MS MAUREEN STILLMAN
301A HARRIS B. DATES DRIVE
ITHACA NY 14850-1313

TEXAS INSTRUMENTS INCORPORATED 1
ATTN: DR DAVID L. WELLS
P.O. BOX 655474, MS 238
DALLAS TX 75265

KESTREL DEVELOPMENT CORPORATION 1
ATTN: DR RICHARD JULLIG
3260 HILLVIEW AVENUE
PALO ALTO CA 94304

DARPA/ITO 1
ATTN: DR KIRSTIE BELLMAN
3701 N FAIRFAX DRIVE
ARLINGTON VA 22203-1714

NASA/JOHNSON SPACE CENTER 1
ATTN: CHRIS CULBERT
MAIL CODE PT4
HOUSTON TX 77058

STERLING IMD INC. 1
KSC OPERATIONS
ATTN: MARK MAGINN
BEECHES TECHNICAL CAMPUS/RT 26 N.
ROME NY 13440

HUGHES SPACE & COMMUNICATIONS 1
ATTN: GERRY BARKSDALE
P. O. BOX 92919
BLDG R11 MS M352
LOS ANGELES, CA 90009-2919

SCHLUMBERGER LABORATORY FOR 1
COMPUTER SCIENCE
ATTN: DR. GUILLERMO ARANGO
8311 NORTH FM620
AUSTIN, TX 78720

DECISION SYSTEMS DEPARTMENT 1
ATTN: PROF WALT SCACCHI
SCHOOL OF BUSINESS
UNIVERSITY OF SOUTHERN CALIFORNIA
LOS ANGELES, CA 90089-1421

SOUTHWEST RESEARCH INSTITUTE 1
ATTN: BRUCE REYNOLDS
6220 CULEBRA ROAD
SAN ANTONIO, TX 78228-0510

NATIONAL INSTITUTE OF STANDARDS 1
AND TECHNOLOGY
ATTN: CHRIS DABROWSKI
ROOM A266, BLDG 225
GAITHSBURG MD 20899

EXPERT SYSTEMS LABORATORY 1
ATTN: STEVEN H. SCHWARTZ
NYNEX SCIENCE & TECHNOLOGY
500 WESTCHESTER AVENUE
WHITE PLAINS NY 20604

NAVAL TRAINING SYSTEMS CENTER 1
ATTN: ROBERT BREAUX/CODE 252
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3224

DR JOHN SALASIN 1
DARPA/ITO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

DR BARRY BOEHM 1
DIR, USC CENTER FOR SW ENGINEERING
COMPUTER SCIENCE DEPT
UNIV OF SOUTHERN CALIFORNIA
LOS ANGELES CA 90089-0781

DR STEVE CROSS 1
CARNEGIE MELLON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
PITTSBURGH PA 15213-3891

DR MARK MAYBURY 1
MITRE CORPORATION
ADVANCED INFO SYS TECH; G041
BURLINTON ROAD, M/S K-329
BEDFORD MA 01730

ISX 1
ATTN: MR. SCOTT FOUSE
4353 PARK TERRACE DRIVE
WESTLAKE VILLAGE,CA 91361

MR GARY EDWARDS 1
ISX
433 PARK TERRACE DRIVE
WESTLAKE VILLAGE CA 91361

DR ED WALKER 1
BBN SYSTEMS & TECH CORPORATION
10 MOULTON STREET
CAMBRIDGE MA 02238

LEE ERMAN 1
CIMFLEX TEKNOWLEDGE
1810 EMBACADERO ROAD
P.O. BOX 10119
PALO ALTO CA 94303

DR. DAVE GUNNING 1
DARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

DAN WELD 1
UNIVERSITY OF WASHINGTON
DEPART OF COMPUTER SCIENCE & ENGIN
BOX 352350
SEATTLE, WA 98195-2350

STEPHEN SODERLAND 1
UNIVERSITY OF WASHINGTON
DEPT OF COMPUTER SCIENCE & ENGIN
BOX 352350
SEATTLE, WA 98195-2350

DR. MICHAEL PITTARELLI 1
COMPUTER SCIENCE DEPART
SUNY INST OF TECH AT UTICA/ROME
P.O. BOX 3050
UTICA, NY 13504-3050

CAPRARO TECHNOLOGIES, INC 1
ATTN: GERARD CAPRARO
311 TURNER ST.
UTICA, NY 13501

USC/ISI 1
ATTN: BOB MCGREGOR
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90292

SRI INTERNATIONAL 1
ATTN: ENRIQUE RUSPINI
333 RAVENSWOOD AVE
MENLO PARK, CA 94025

DARTMOUTH COLLEGE 1
ATTN: DANIELA RUS
DEPT OF COMPUTER SCIENCE
11 ROPE FERRY ROAD
HANOVER, NH 03755-3510

UNIVERSITY OF FLORIDA 1
ATTN: ERIC HANSON
CISE DEPT 456 CSE
GAINESVILLE, FL 32611-6120

CARNEGIE MELLON UNIVERSITY 1
ATTN: TOM MITCHELL
COMPUTER SCIENCE DEPARTMENT
PITTSBURGH, PA 15213-3890

CARNEGIE MELLON UNIVERSITY 1
ATTN: MARK CRAVEN
COMPUTER SCIENCE DEPARTMENT
PITTSBURGH, PA 15213-3890

UNIVERSITY OF ROCHESTER 1
ATTN: JAMES ALLEN
DEPARTMENT OF COMPUTER SCIENCE
ROCHESTER, NY 14627

TEXTWISE, LLC 1
ATTN: LIZ LIDDY
2-121 CENTER FOR SCIENCE & TECH
SYRACUSE, NY 13244

WRIGHT STATE UNIVERSITY 1
ATTN: DR. BRUCE BERRA
DEPART OF COMPUTER SCIENCE & ENGIN
DAYTON, OHIO 45435-0001

UNIVERSITY OF FLORIDA 1
ATTN: SHARMA CHAKRAVARTHY
COMPUTER & INFOR SCIENCE DEPART
GAINESVILLE, FL 32622-6125

KESTREL INSTITUTE 1
ATTN: DAVID ESPINOSA
3260 HILLVIEW AVENUE
PALO ALTO, CA 94304

USC/INFORMATION SCIENCE INSTITUTE 1
ATTN: DR. CARL KESSELMAN
11474 ADMIRALTY WAY, SUITE 1001
MARINA DEL REY, CA 90292

MASSACHUSETTS INSTITUTE OF TECH 1
ATTN: DR. MICHAELE SIEGEL
SLOAN SCHOOL
77 MASSACHUSETTS AVENUE
CAMBRIDGE, MA 02139

USC/INFORMATION SCIENCE INSTITUTE 1
ATTN: DR. WILLIAM SWARTHOUT
11474 ADMIRALTY WAY, SUITE 1001
MARINA DEL REY, CA 90292

STANFORD UNIVERSITY 1
ATTN: DR. GIO WIEDERHOLD
857 SIERRA STREET
STANFORD
SANTA CLARA COUNTY, CA 94305-4125

NCCOSC RDTE DIV D44208 1
ATTN: LEAH WONG
53245 PATTERSON ROAD
SAN DIEGO, CA 92152-7151

SPAWAR SYSTEM CENTER 1
ATTN: LES ANDERSON
271 CATALINA BLVD, CODE 413
SAN DIEGO CA 92151

GEORGE MASON UNIVERSITY 1
ATTN: SUSHIL JAJODIA
ISSE DEPT
FAIRFAX, VA 22030-4444

DIRNSA 1
ATTN: MICHAEL R. WARE
DOD, NSA/CSS (R23)
FT. GEORGE G. MEADE MD 20755-6000

DR. JIM RICHARDSON 1
3660 TECHNOLOGY DRIVE
MINNEAPOLIS, MN 55418

LOUISIANA STATE UNIVERSITY 1
COMPUTER SCIENCE DEPT
ATTN: DR. PETER CHEN
257 COATES HALL
BATON ROUGE, LA 70803

INSTITUTE OF TECH DEPT OF COMP SCI 1
ATTN: DR. JAIDEEP SRIVASTAVA
4-192 EE/CS
200 UNION ST SE
MINNEAPOLIS, MN 55455

GTE/BBN 1
ATTN: MAURICE M. MCNEIL
9655 GRANITE RIDGE DRIVE
SUITE 245
SAN DIEGO, CA 92123

UNIVERSITY OF FLORIDA 1
ATTN: DR. SHARMA CHAKRAVARTHY
E470 CSE BUILDING
GAINESVILLE, FL 32611-6125

AFRL/IFT 1
525 BROOKS ROAD
ROME, NY 13441-4505

AFRL/IFTM 1
525 BROOKS ROAD
ROME, NY 13441-4505